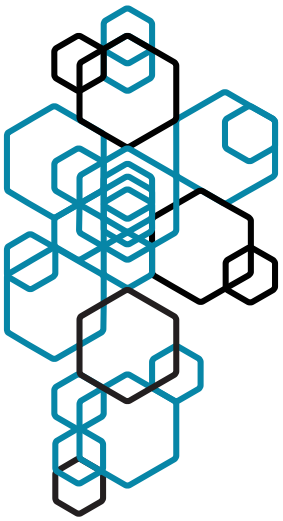
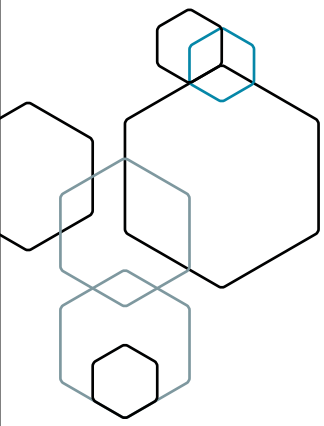
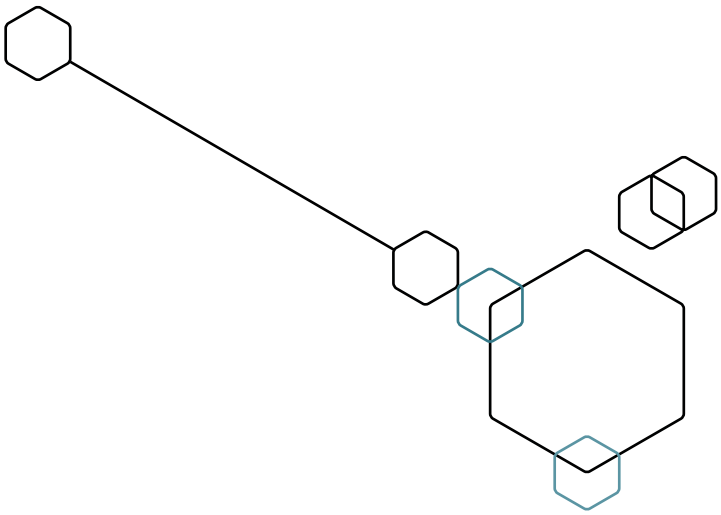
ICMC 2016

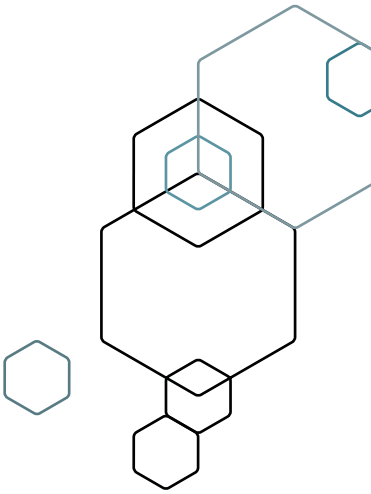
42nd International Computer Music Conference
12th –16th September 2016
Utrecht, The Netherlands

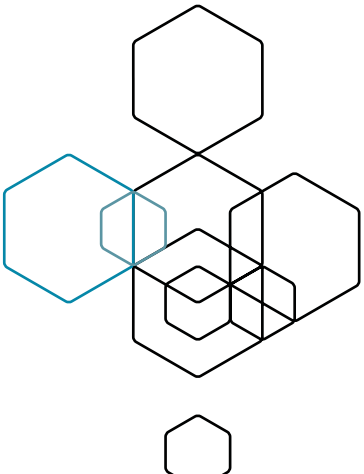




Proceedings

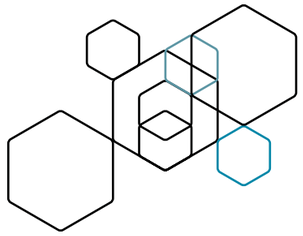






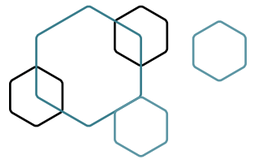
ICMC 2016

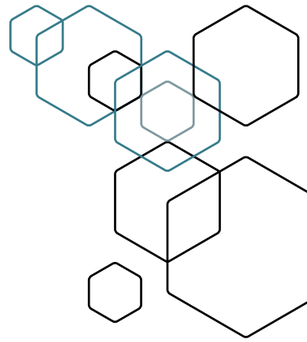
www.icmc2016.com



Is the sky the limit?

Utrecht, 12-16 September 2016





Proceedings

hosted by

HKU University of the Arts Utrecht, HKU Music and Technology

and Gaudeamus Muziekweek

Proceedings of the 42st International Computer Music Conference
12-16 September

HKU University of the Arts Utrecht, HKU Music and Technology
Gaudeamus Muziekweek

Is the sky the limit?

Hans Timmermans, editor.

ISBN-10: 0-9845274-5-1 ISBN-13: 978-0-9845274-5-8

Published by:
HKU University of the Arts Utrecht, HKU Music and Technology
Ina Boudier-Bakkerlaan 50
3582 VA Utrecht
The Netherlands

The ICMC-2016 is supported by:

Welcome notes
Organisation

A most appropriate background

The International Computer Music Association has chosen Utrecht for its 42nd Conference. So welcome to our city, where the music of the Venetian School still resonates after the Early Music Festival, which ended a week ago. And where the newest music, presented by the best young composers and performing artists, is still fresh in the mind after the Gaudeamus Muziekweek.

The ICMA’s choice of Utrecht seems particularly apt, since our city is home not only to art, but also to the second pillar of computer music – science. The seven faculties of our university, the country’s best, are authoritative in their fields, such as geosciences, life sciences, and humanities. One fifth of our population consists of students, a demographic factor that contributes greatly to the atmosphere of the city.

Along with many other institutions and festivals, Gaudeamus and HKU University of the Arts are staunch representatives of our city’s strong cultural profile, as is our new, all-round TivoliVredenburg concert hall.

In short, Utrecht provides a most appropriate background to your conference, and will not fail to inspire you. Please enjoy your stay, don’t forget to look around (and listen, of course) and keep Utrecht in mind when you’re next planning a city trip with family and friends (musical or otherwise).

Jan van Zanen

Mayor of Utrecht

Dear visitor of the ICMC 2016,

It gives us great pleasure to welcome you to the ICMC 2016, the 42nd edition of the International Computer Music Conference.
The main question of the conference “Is the sky the limit?” will be explored through a variety of concerts, installations, paper presentations, workshops, installations and events, by composers, researchers, sonic artists, students, sound designers, professors and many others. It will be an intense week in the city of Utrecht, which is known as the musical centre of the Netherlands and is home to institutions like Gaudeamus Muziekweek and HKU Music and Technology, the two institutions that have collaboration closely in organising the ICMC 2016.
The main venue of the conference is TivoliVredenburg, housing two major concert halls, three smaller halls and many open spaces for meeting and networking with your peers. The traditional ‘day out and banquet’ will be held in the botanical gardens of Utrecht University, one of the most beautiful of its kind in the Netherlands, allowing you to relax after intense discussions and exciting concerts.
All in all, we hope the programme will be interesting and challenging and we wish you an adventurous and inspiring week in every respect!

Conference chairs
Henk Heuvelmans, director of Gaudeamus Muziekweek
Rens Machielse, director of Music and Technology at HKU University of the Arts Utrecht

Dear 2016 ICMC Delegates,

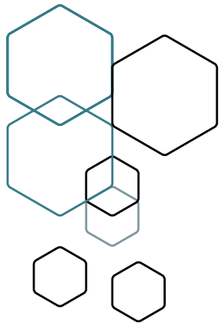
I am very happy to welcome you to ICMC 2016, the 42nd International Computer Music Conference hosted by HKU University of the Arts Utrecht and Gaudeamus Muziekweek. I am excited to be here in Utrecht, an important city for computer music research and innovative music. It is the birthplace of Louis Andriessen and Koenig and Berg’s groundbreaking SSP music language, and home to the Institute of Sonology, as well as our hosts: the Gaudeamus Foundation and HKU University of the Arts Utrecht.

The theme of this conference poses the question “Is the sky the limit?” Now that the field of Computer Music is well over fifty years old, it is very appropriate to reflect on this theme. Computer hardware is now cheap and fast, music distribution is practically free, and the ideas of computer music have spread far beyond the original genre of Computer Music. This conference can inspire us to consider bigger ideas, to revisit techniques previously thought impractical, and to expand our work into a broader musical community.

I would like to thank the hosts of this conference for all their technical and aesthetic guidance: Martijn Buser, Hans Timmermans, Henk Heuvelmans and Rens Machielse. We have a lot to look forward to: An intriguing keynote from composer/ performer/artist Åke Parmerud, special off-ICMC performances by Tarik Barri, Thomas Ankersmit, Taraf, Allert Alders and Robert Henke, and a full schedule of paper sessions, concerts, installations and workshops featuring our own work. I congratulate our hosts for organising a wonderful week of music, research, inspiration and good company.

Welcome to the 2016 International Computer Music Conference!

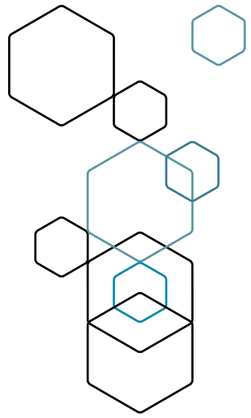
Tom Erbe, ICMA President



Welcome to Utrecht!

Utrecht has an amazing infrastructure of venues located in the city centre. The eye-catching TivoliVredenburg (with no fewer than 5 venues!) is a unique building that is ideally suited to hosting this year's International Computer Music Conference. We are very proud to present over 100 compositions and 6 installations, in 16 concerts, 4 listening rooms and 6 off-ICMC events. I would like to thank over a hundred reviewers for doing the difficult job of reviewing nearly 600 submissions.
We hope that the selection moves you and 'feeds' your creativity.

Martijn Buser, Music/Listening Room Chair



Welcome from the Paper Chair

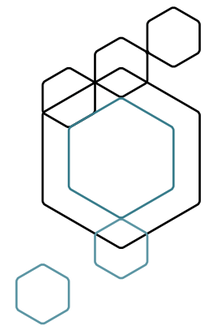
We are happy to welcome you to the 2016 International Computer Music Conference and to the city of Utrecht. Utrecht has a long history of computer music, as the Institute of Sonology was founded at Utrecht University in 1960, where many of us studied or worked, or at least discovered computer music.
In 1986, the Institute of Sonology moved to the Royal Conservatory of The Hague, which was hosting that year's International Computer Music Conference, 30 years ago. In fact, it was my first - and certainly not my last - ICMC. The HKU Music and Technology programme was founded in 1985, and we are pleased to be collaborating with Gaudeamus Muziekweek in organising the ICMC 2016.

We are proud to present the proceedings of ICMC 2016. We received a total of 160 paper submissions from 28 countries, of which 119 submissions were accepted and scheduled.

The submissions were reviewed using a double-blind review process, and each submission received around three conscientious and often quite detailed reviews. This year's review committee was comprised of 112 reviewers from 20 countries, representing a wide spectrum of specialisations in the computer music field. This year we accepted 40 long papers, 46 short papers, 27 posters/demos and 6 workshop proposals. Reviewing and adjudicating the many high-quality submissions is never easy, and we had to take some difficult decisions. We are sorry that some of the accepted papers could not be presented by one of the authors and had to be removed from the programme for that reason. We feel that the selected papers strongly represent the current research, development and aesthetic thought in computer music today.

We wish you a very inspiring ICMC2016 @ UTRECHT !

Hans Timmermans
Paper Chair, ICMC 2016



Paper selection committee and reviewers

Full Name	Organization	Country
Miriam Akkermann	Bayreuth University	Germany
Jesse Allison	Louisiana State University	United States
Georgaki Anastasia	UNIVERSITY OF ATHENS	Greece
Torsten Anders	University of Bedfordshire	United Kingdom
Ted Apel		United States
Mark Ballora	Penn State University	United States
Leah Barclay	Griffith University	Australia
Natasha Barrett	University of Oslo, Department for Musicology.	Norway
Stephen David Beck	Louisiana State University	United States
Peter Beyls	CITAR - UCP	Portugal
Jamie Bullock	Birmingham City University	United Kingdom
John Ashley Burgoyne	Universiteit van Amsterdam	Netherlands
Christopher Burns	University of Michigan	United States
Juan Jose Burred		France
Baptiste Caramiaux	IRCAM / McGill University	Canada
Nicolas Castagné	Grenoble INP - ICA laboratory - ACROE	France
Chris Chafe	CCRMA Stanford University	United States
Marko Ciciliani	University of Music and Performing Arts Graz/IEM	Austria
David Coll	Freelance Composer & Sound Artist	United States
David Cope	UC Santa Cruz	United States
Cathy Cox	Kunitachi College of Music	Japan
Roger Dannenberg	Carnegie Mellon University	United States
Giovanni De Poli	DEI - University of Padova	Italy
Symeon Delikaris-Manias	Aalto University	Finland
Paul Doornbusch	Australian College of the Arts	Australia
Richard Dudas	Hanyang University	South Korea
Aaron Einbond	City University London	United Kingdom
Tom Erbe	UCSD	United States
Cumhur Erkut	Aalborg University Copenhagen	Denmark
Georg Essl	University of Michigan	United States
Carl Faia	Brunel University London	United Kingdom
John ffitch		United Kingdom
Rebecca Fiebrink	Goldsmiths University of London	United Kingdom
Rajmil Fischman	Keele University	United Kingdom
Dominique Fober	Grame	France
Ivan Franco	McGill University	Canada
Pete Furniss	University of Edinburgh (ECA)	United Kingdom
Dr. Gregorio García Karman	Akademie der Künste	Germany
Michael Gatt	Kingston University London	United Kingdom
Jean-Louis Giavitto	IRCAM - CNRS - Inria	France

Full Name	Organization	Country
Mick Grierson	Goldsmiths	United Kingdom
Michael Gurevich	University of Michigan	United States
Rob Hamilton	Rensselaer Polytechnic Institute	United States
Ian Hattwick	McGill University	Canada
Christopher Haworth	University of Oxford	United Kingdom
Lauren Hayes	Arizona State University	United Kingdom
Mara Helmuth	University of Cincinnati	United States
Henk Heuvelmans	Gaudeamus Muziekweek	Netherlands
Jason Hockman	Birmingham City University	United Kingdom
Alexander Refsum Jensenius	University of Oslo	Norway
Jean Marc Jot	DTS, Inc.	United States
Emmanuel Jourdan	Ircam	France
Steven Kemper	Mason Gross School of the Arts, Rutgers University	United States
David Kim-Boyle	University of Sydney	Australia
Michèl Koenders	HKU University of the Arts Utrecht	Netherlands
Juraj Kojs	University of Miami	United States
Johnathan F. Lee	Tamagawa University	Japan
Serge Lemouton	ircam	France
PerMagnus Lindborg, PhD	Nanyang Technological University	Singapore
Cort Lippe	University of Buffalo	United States
Eric Lyon	Virginia Tech	United States
John MacCallum	CNMAT / UC Berkeley	United States
Rens Machielse	HKU University of the Arts Utrecht	Netherlands
Thor Magnusson	University of Sussex	United Kingdom
Joseph Malloch	Inria	France
Mikhail Malt	IRCAM	France
Peter Manning	Durham University, UK	United Kingdom
Cory McKay	Marianopolis College	Canada
Andrew McPherson	Queen Mary University of London	United Kingdom
David Medine	University of California, San Diego	United States
Christos Michalakos	University of Abertay	United Kingdom
Nicolas MISDARIIS	STMS Ircam-CNRS-UPMC	France
Peter Nelson	University of Edinburgh	United Kingdom
Jérôme Nika	Ircam	France
Reid Oda	Princeton University, Department of Computer Science	United States
Erik Oña	Eslektronische Studio Basel, Musikhochschulen FHNW, Musikakademie Basel	Switzerland
Timothy Opie	Box Hill Institute	Australia
Miguel Ortiz	Goldsmiths, University of London	United Kingdom
Naotoshi Osaka	Tokyo Denki University	Japan
Laurent Pottier	université de Saint-Etienne	France
Miller Puckette	UCSD	United States
Curtis Roads	UCSB	United States
Ilya Rostovtsev	CNMAT, UC Berkeley	United States
Robert Rowe	New York University	United States
Jøran Rudi	NOTAM	Norway
Adriana Sa	EAVI / Goldsmiths, University of London	Portugal

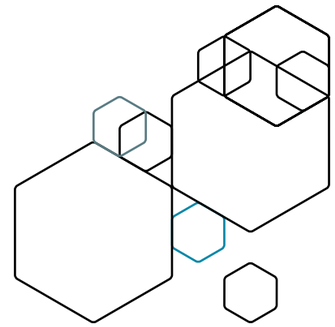
Full Name	Organization	Country
Diana Salazar	Royal Conservatoire of Scotland	United Kingdom
Mihir Sarkar	Musikara, Inc.	United States
Carla Scaletti	Symbolic Sound	United States
margaret schedel	Stony Brook University	United States
Diemo Schwarz	Ircam	France
Alexander Sigman	International College of Liberal Arts (iCLA), Yamanashi Gakuin University	Japan
Stephen Sinclair	INRIA Chile	Chile
Mattias Sköld	Royal College of Music in Stockholm	Sweden
Benjamin Smith	Indiana University-Purdue University-Indianapolis	United States
Tamara Smyth	UCSD	United States
Andrew Sorensen		Australia
Hans Timmermans	HKU - Utrecht university of the Arts	Netherlands
George Tzanetakis	University of Victoria	Canada
Rafael Valle	Center for New Music and Audio Technologies	United States
Doug Van Nort		
Lindsay Vickery	Edith Cowan University	Australia
Graham Wakefield	York University	Canada
Johnty Wang	Input Devices and Music Interaction Laboratory, McGill University	Canada
Simon Waters	Sonic Arts Research Centre, Queen’s University Belfast	United Kingdom
Andreas Weixler	CMS - Brukcner University, Linz	Austria
Marcel Wierckx	HKU - Utrecht university of the Arts	Netherlands
Matthew Yee-King	Goldsmiths	United Kingdom
Sølvi Ystad	LMA-CNRS	France
Michael Zbyszynski	Goldsmiths University of London	United Kingdom

Music selection committee and reviewers

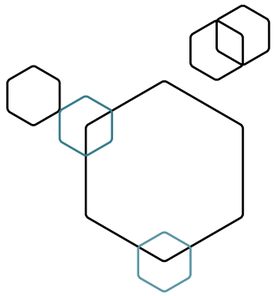
Full Name	Organization	Country
Armeno Alberts	Stichting CEM / Concertzender / independent composer - musician	Netherlands
James Andean	De Montfort University	United Kingdom
Idske Bakker	Insomnio	Netherlands
Claudio F Baroni		Netherlands
Natasha Barrett	University of Oslo, Department for Musicology.	Norway
Nicolas Bernier	Université de Montréal	Canada
Ivo Bol		Netherlands
Peter + Simone Bosch + Simons	Bosch & Simons	Spain
Martijn Buser	Gaudeamus Muziekweek	Netherlands
Chris Chafe	CCRMA / Stanford	United States
Se-Lien Chuang	Atelier Avant Austria	Austria
Marko Ciciliani	University of Music and Performing Arts Graz/IEM	Austria
Ricardo Climent	NOVARS Research Centre, University of Manchester	United Kingdom
Agostino Di Scipio		Italy
Ingrid DRESE	Arts2_Conservatoire Royale de Musique de Mons	Belgium
Richard Dudas	Hanyang University	South Korea
Christian Eloy	SCRIME - Université Bordeaux 1	France
Antonio Ferreira		Portugal
Jason Freeman	Georgia Institute of Technology	United States
Douglas Geers	City University of New York	United States
Carlos Guedes	New York University Abu Dhabi	United Arab Emirates
Jonty Harrison	University of Birmingham (retired)	United Kingdom
Mara Helmuth	University of Cincinnati	United States
Henk Heuvelmans	Gaudeamus Muziekweek	Netherlands
Rozalie Hirs	Rozalie Hirs	Netherlands
Christopher Hopkins	Iowa State University of Science and Technology	United States
Luc Houtkamp		Malta
Guy van Hulst	TivoliVredenburg	Netherlands
Shintaro Imai	Kunitachi College of Music	Japan
Vera Ivanova	Chapman University/Colburn School	United States
Orestis Karamanlis	Bournemouth University	United Kingdom
Konstantinos Karathanasis	University of Oklahoma, School of Music	United States
Bronne Keesmaat	Rewire	Netherlands
David Kim-Boyle	Sydney Conservatorium of Music, University of Sydney	Australia
Juraj Kojs	University of Miami	United States
Panayiotis Kokoras	University of North Texas	United States

Full Name	Organization	Country
Paul Koonce	University of Florida	United States
Yannis Kyriakides		Netherlands
Anne La Berge	Volsap Foundation	Netherlands
Lin-Ni LIAO	IReMus	France
Cort Lippe	University of Buffalo	United States
Apostolos Loufopoulos	Ionian University, Department of Audio&Visual Arts, assistant professor	Greece
Minjie LU	Sichuan Conservatory of Music, China	China
Stelios Manousakis	Stichting Modulus	Netherlands
Mario MARY	Academy Rainier III	Monaco
Ezequiel Menalled	Ensemble Modelo62	Netherlands
Scott Miller	St Cloud State University	United States
Marco Momi		Italy
Hugo Morales Murguia		Netherlands
Jon Nelson	University of North Texas, CEMI	United States
Vassos Nicolaou		Germany
Erik Nystrom	University of Birmingham	United Kingdom
Kjartan Olafsson	ErkiTónlist sf - IAA	Iceland
Joao Pedro Oliveira	Federal University of Minas Gerais	Brazil
christisna oorebeek	autonomous composer	Netherlands
Felipe Otondo	Universidad Austral	Chile
Gabriel Paiuk	Institute of Sonology - Royal Conservatoire The Hague	Netherlands
Tae Hong Park	New York University	United States
Åke Parmerud		Sweden
Juan Parra	Orpheus Institute, Ghent	Belgium
Rui Penha	INESC TEC / FEUP	Portugal
Ulrich Pöhl	Insomnio	Netherlands
Michal Rataj	Academy Of Performing Arts, Prague	Czech Republic
Michael Rhoades	The Perception Factory	United States
Sebastian Rivas		France
Manuel Rocha Iturbide	Universidad Autónoma Metropolitana	Mexico
Erwin Roebroeks	Erwin Roebroeks	Netherlands
Margaret Schedel	Stony Brook University	United States
Federico Schumacher	Universidad Diego Portales	Chile
Wouter Snoei		Netherlands
Antonio Sousa Dias		Portugal
Roland Spekle	HKU	Netherlands
Georgia Spiropoulos		France
Kurt Stallmann	Shepherd School of Music, Rice University	United States
Adam Stansbie	The University of Sheffield	United Kingdom
Nikos Stavropoulos	Leeds Beckett University	United Kingdom
Pete Stollery	University of Aberdeen	United Kingdom
Jeroen Strijbos	Strijbos & Van Rijswijk	Netherlands
Martin Supper	Berlin University of the Arts	Germany
Jorrit Tamminga	Conservatorium van Amsterdam	Netherlands
Kees Tazelaar	Institute of Sonology / Royal Conservatoire	Netherlands
Jacob Ter Veldhuis	Boombox	Netherlands
Robert Scott Thompson	Georgia State University	United States

Full Name	Organization	Country
Hans Timmermans	HKU - Utrecht university of the Arts, Music and Technology	Netherlands
Pierre Alexandre Tremblay	University of Huddersfield	United Kingdom
Daniel Trueman	Princeton University	United States
Yu Chung Tseng	National Chiao Tung University in Taiwan	Taiwan
Anders Tveit		Norway
Katerina Tzedaki	Department of Music Technology & Acoustics Engineering, Technological Educational Institute of Crete	Greece
René Uijlenhoet	Codarts	Netherlands
Peter van Bergen	LOOS Foundation	Netherlands
Lucas van der Velden	Sonic Acts	Netherlands
Cathy van Eck	Bern University of the Arts	Switzerland
Robert van Heumen		Netherlands
Rob van Rijswijk	Strijbos & Van Rijswijk	Netherlands
Annette Vande Gorne	Musiques & Recherches	Belgium
Henry Vega	ARTEk Foundation	Netherlands
Rodney Waschka	North Carolina State University	United States
Andreas Weixler	CMS Computer Music Studio, Bruckner University Linz	Austria
Daniel Weymouth	SUNY Stony Brook	United States
Marcel Wierckx		Netherlands
XIAO FU ZHANG	Electroacoustic Music Association of China, Central Conservatory of Music	China
Lidia Zielinska	Paderewski Academy of Music	Poland



ICMA Paper Awards



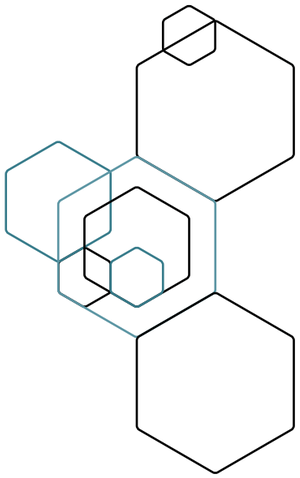
Every year, the ICMA presents the Best Paper Award to the best paper submitted. Papers with the highest score, written by ICMA members, are given to a panel elected by the ICMA Board, who decide on a winner. And at the end of the conference, attendees cast their votes for the winner of the Best Presentation Award . The winner of the award is announced in the proceedings for the following year. Look out for the ballot box and cast your vote at this year's conference!!

ICMC 2016 Best Paper Award

Lauren Hayes

- for -

Sound, Electronics and Music: an evaluation of early embodied education



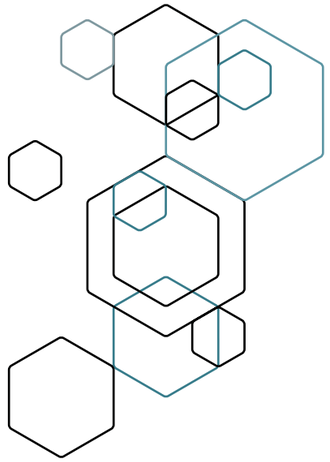
2016 Paper Award Panel:
Rebecca Fiebrink, Chair
Meg Schedel
Stefania Serafin
Tae Hong Park
Matthew Blessing

ICMC 2015 Best Paper Award

Greg Surges, Tamara Smyth & Miller Puckette

- for -

Generative Feedback Networks Using Time-Varying Allpass Filters



ICMC 2015 Best Presentation Award:

Dekai Wu and Karteek Addanki

- for -

Neural Versus Symbolic Rap Battle Bots

ICMC 2014 Best Presentation Award

Christopher Trapani & José Echeveste

- for -

Real Time Tempo Canons with Antescofo

ICMC 2013 Best Presentation Award

Lonce Wyse & Pallav Shinghal

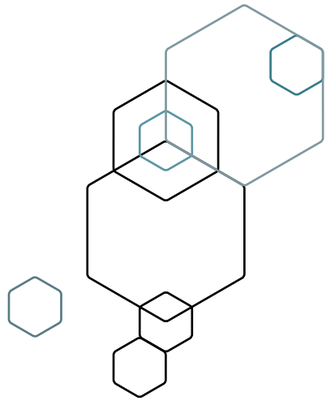
- for -

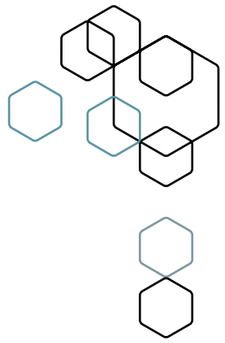
Sonicbard: Storytelling With Real-Time Sound Control,Synthesis and Processing Using Emerging Browser-Based Technologies

The ICMA Music Awards 2016 are as follows:

Europe: Ricardo Climent, for slaag
Asia and Oceania: Hongshuo Fan, for Extrema
Americas: Rob Hamilton and Chris Platz, for Carillon
Student: Sang Won Lee, for Live Writing : Gloomy Streets

This year's ICMA music awards committee was coordinated by PerMagnus Lindborg and comprised Christopher Haworth, Chryssie Nanou and Eric Honour, receiving additional input from Miriam Akkermann, Charles Nichols and John Thompson. The shortlist of forty works contained many strong candidates and the jury's task was not an easy one. Committee members independently evaluated the artistic and technical merits of each work, and our final decision was reached through discussion and careful deliberation. We were thoroughly impressed by the high overall standard, and would like to extend our warmest congratulations to the winners.





Organising team

Conference Chairs:
Rens Machielse, director of HKU University of the Arts Utrecht Music and Technology, and
Henk Heuvelmans, director of Gaudeamus Muziekweek

Paper Chair:
Hans Timmermans, senior lecturer/researcher at HKU University of the Arts Utrecht Music and Technology

Music/Listening Room Chair:
Martijn Buser, programmer of Gaudeamus Muziekweek

Off ICMC Chair:
Roland Spekle, fellow at HKU University of the Arts Utrecht Music and Technology

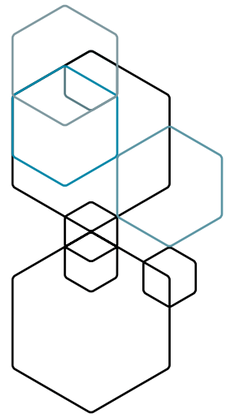
Technical team:
Elizabet van der Kooij (chair), Thomas Koopmans and Poul Holleman

Administration/coordination:
Gaudeamus and HKU University of the Arts Utrecht

Press:
Laura Renard and Femke Langebaerd

Projectmanager:
Tamara Kalf

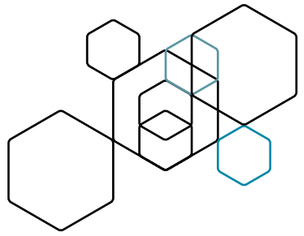
Design:
Saskia Freeke



ICMA Board of Directors

ICMA Officers

President	Tom Erbe
Vice President for Membership	Michael Gurevich
Vice President for Conferences	Margaret Schedel
Vice President for Asia/Oceania	Lonce Wyse
Vice President for Americas	Madelyn Byrne
Vice President for Europe	Stefania Serafin
Vice President for Preservation	Tae Hong Park
Treasurer/Secretary	Chryssie Nanou
Publications Coordinator	Rob Hamilton
Research Coordinator	Rebecca Fiebrink
Music Coordinator	PerMagnus Lindborg
Array Editor	Christopher Haworth

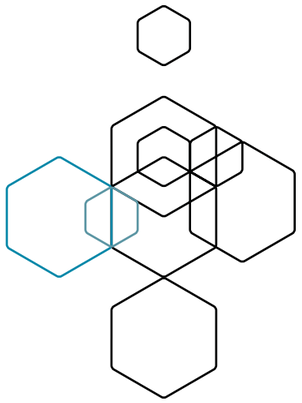


ICMA Board of Directors 2016

At-Large Directors	Miriam Akkermann
	Tom Erbe
	Mark Ballora
	John Thompson
Oceania Regional Directors	Takeyoshi Mori
	Lonce Wyse
Europe Regional Directors	Stefania Serafin
	Arshia Cont

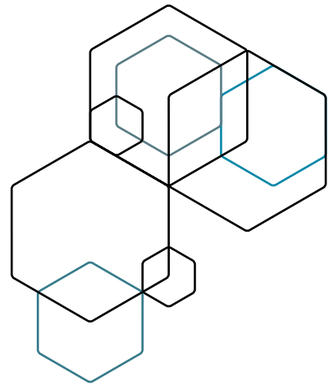
Non-elected officers

ICMA Administrative Assistant	Sandra Neal

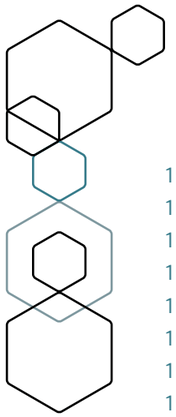


List of Previous Conferences

2015	Texas, USA
2014	Athens, Greece
2013	Perth, Australia
2012	Ljubljana, Slovenia
2011	Huddersfield, England, UK
2010	New York City, New York, USA
2009	Montreal, Quebec, Canada
2008	Belfast, N. Ireland, UK
2007	Copenhagen, Denmark
2006	New Orleans, Louisiana, USA
2005	Barcelona, Spain
2004	Miami, USA
2003	Singapore
2002	Gothenburg, Sweden
2001	Havana, Cuba
2000	Berlin, Germany



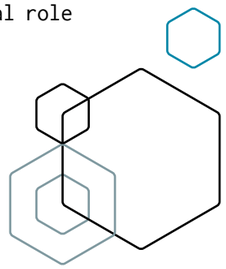
1999	Beijing, China
1998	Ann Arbor, Michigan, USA
1997	Thessaloniki, Greece
1996	Hong Kong, China
1995	Banff, Alberta, Canada
1994	Aarhus, Denmark
1993	Tokyo, Japan
1992	San Jose, California, USA
1991	Montreal, Quebec, Canada
1990	Glasgow, Scotland, UK



1989	Columbus, Ohio, USA
1988	Cologne, Germany
1987	Champaign/Urbana, Illinois, USA
1986	Den Haag, Netherlands
1985	Burnaby, British Columbia, Canada
1984	Paris, France
1983	Rochester, New York, USA
1982	Venice, Italy
1981	Denton, Texas, USA
1980	New York City, New York, USA
1978	Chicago, Illinois, USA
1977	San Diego, California, USA
1976	Cambridge, Massachusetts, USA
1975	Champaign/Urbana, Illinois, USA
1974	East Lansing, Michigan, USA

Conference themes

The main theme of ICMC 2016 will be “Is the sky the limit?”. This theme is divided into five sub-themes, each of which will play a central role on each conference day.



1. Is the sky determined by technology or aesthetics?

The creative process and the associated aesthetics in electronic music have always been largely defined by technology. This technology has now been developed to such an extent that often it is no longer seen as a defining and/or restricting element.

2. Is the sky local?

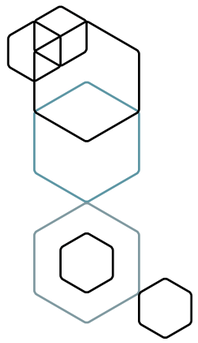
Innovation starts on a small scale – in labs and educational institutes, and through visionary individuals. What they have in common is their place at the foundation of initiatives that aim to radically change the course of music history.

3. Educating for the sky

Courses in Computer Music and/or Music Technology come in all sorts of gradations and cultural views. The question might be – what is our educational goal and what is advisable? The answer to this question will be strongly influenced by different contexts.

4. Does the sky need a composer or musician?

Do we still need musicians? Or composers? Why do we need an audience? Is there no direct distribution to your audience? Has electronic or computer music become a logical or natural part of contemporary music?

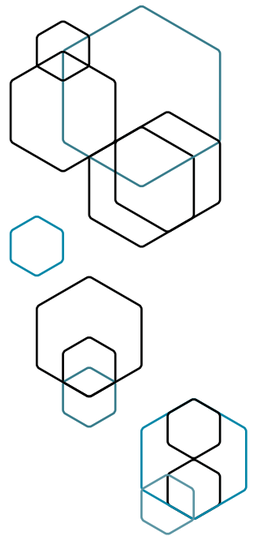


5. Stretching the sky

Since the 1950’s, new music has alienated itself completely from society and now only operates within an almost immeasurably small niche of enthusiasts. The general public has no knowledge or love of it. While electronic music originally played a small part, on closer inspection it has developed much more broadly and is increasingly present in all sorts of layers of society, from musical contexts to social sectors, from concert halls to healthcare, and from the public domain to computer games.

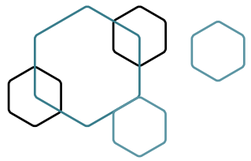
1. Is the sky determined by technology or aesthetics?

The creative process and the associated aesthetics in electronic music have always been largely defined by technology. This technology has now been developed to such an extent that often it is no longer seen as a defining and/or restricting element. It is, however, the question whether this is justified. The development of specific interfaces in music technology applications has an indirect influence on the user’s behaviour – and therefore also on his or her musical choices. So it is important to consider whether we really do not experience restrictions from technology any longer in the creative process; restrictions that we might want to remove with the assistance of new technology that has less influence on the process. The underlying question is what music we would then like to make that we cannot make at the moment. What would it be like if the imaginative powers of music and the associated idiom and grammar were to define the design of technology? Or can we actually make ‘everything’ already, whether or not with the occasional technological detour? Or is this complete nonsense and are we only at the beginning of, for example, new forms of interaction between the performer and an electronic instrument, which are many times more complex than we can now imagine? Are there still very different sounds and sound structures conceivable, which require another form of technology and other forms of interaction with that technology? And could those other forms lead to new creative processes and new aesthetics?

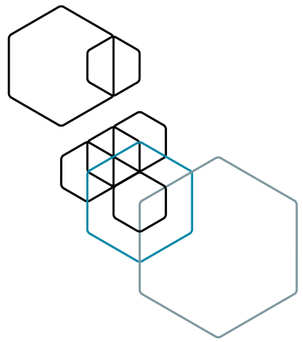


2. Is the sky local?

Innovation starts on a small scale – in labs and educational institutes, and through visionary individuals. What they have in common is their place at the foundation of initiatives that aim to radically change the course of music history. Not every idea or experiment reaches the wider public. Many disappear off the radar somewhere between concept and end product. Just by looking at the role of electronic music in our society, it is an incontestable fact that there is a huge contribution from educational institutes and the innovative technological sector. A visionary is characterised by his ability to see beyond the horizon. And it is typical of a start-up that it aims to distribute the visionary's product worldwide. The wish to give a new concept the widest possible reach is what motivates nearly all makers and producers. Electronic music began in a small number of studios. The most popular type, reflected in the dance culture, is big business and spread all over the world. Also in advertising, films and studios (to name but a few examples), it is unimaginable that no use should be made of technology developed in laboratories.

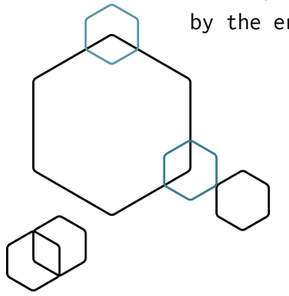


However, reaching a wide public is by no means the only yardstick of the quality of innovation. In the first place, innovation in computer music is approached from a technological perspective. In a changing society, innovation can also be understood to mean the application of products in different contexts and cultures. The essence of innovation, after all, is to break with existing rules and develop new ones. The economic crisis, climate change and other determining developments have a great influence on our ideas about the concept of innovation. Concepts like sustainability and social responsibility play an important role in the cultural and technological debate. Especially for ICMC 2016, HKU will develop a project in a working-class district of Utrecht, where composers and (sound) artists will take up temporary residence, in order to develop projects together with local residents. One interesting aspect of these districts is that they are usually multicultural. The principle behind Is the Sky Local? is the challenge of becoming embedded in the society at grass roots level and engaging with residents to produce original and often multicultural projects.



3. Educating for the sky

Courses in Computer Music and/or Music Technology come in all sorts of gradations and cultural views. In Europe, for example, we have courses with a strong artistic focus, a strong focus on technology, a focus on personal development, a focus on research and a strong focus on existing professional practice. The question might be – what is our educational goal and what is advisable? The answer to this question will be strongly influenced by the culture in which the question is answered, by the institute, by the background of whoever is formulating the answer, by legislation, tradition and other customs, and by the era in which this question is answered.



There are contexts in which tradition dictates that the student's artistic development has top priority. In Europe, however, higher education is increasingly judged on the extent to which it links up with existing professional practice.

These two perspectives on "Educating for the sky" appear to contradict one another (depending on the definition of existing professional practice). It may be wiser to answer the question by charting existing situations along with the relevant arguments.

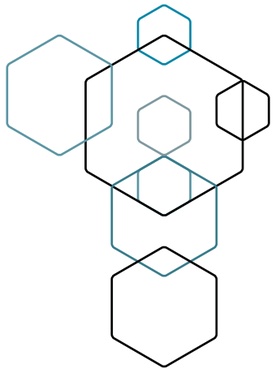
This inventory may lead to mutual comparisons and to greater understanding of what "Educating for the sky" could mean and how courses in Computer Music and Music Technology have developed in the recent past and will develop in the near future.

4. Does the sky need a composer or musician?

Do we still need musicians? Or composers? Why do we need an audience? Is there no direct distribution to your audience? Has electronic or computer music become a logical or natural part of contemporary music?

The role of the maker

To an increasing extent, all we hear around us nowadays is electronic music – in all sorts of media, in the public domain, in clubs and on stage. Take the success of the Dutch dance industry, for example. At the same time, technology makes it possible to make your own electronic music and tracks – with minimal knowledge of music and technology. You don't even need to have learned how to play an instrument. You can compose something on your iPad in a trice and share it with your friends a few seconds later on Soundcloud. Everyone has become a composer and musician! And we can close down the music academies. What right have you still got to call yourself a composer? Is it necessary to call yourself that at all? And what audience is there then? Only consumers (e.g. the dancing crowds at Sensation White, etc.)? Or is everyone a prosumer nowadays? And what is the role of 'performance' in electronic music? Maybe that is precisely what gives you the opportunity (playing an instrument) to distinguish yourself from all those other 'composers'. But what are the essential elements of the 'performance of electronic music'? Is this discussion limited to electronic music anyway, or is it just a discussion about music in general?



Digital music

Furthermore, there is an enormous amount of music archived on the internet. So why should you still want to go to a live concert? What is it that draws an audience to a live performance that they can't get at home on their laptop? Is it a musician after all? Or an author/transmitter? Sharing your love of music doesn't necessarily have to take place in the concert hall. It is actually even easier to share your love for the work online. And why do we still need stages and festivals if all music can be digitally distributed? Maybe we are moving towards a world in which stages and festivals will exist in the form of social networks and are no longer a physical place where people come together. What effect will this have on our understanding and experience of music? How will it affect the practice of music-making and performing? And what is the role of new technologies in this process?

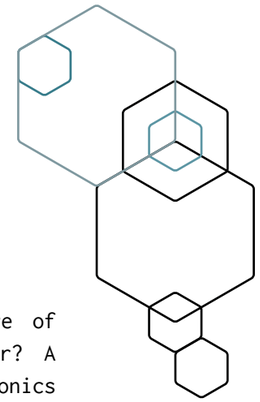
5. Stretching the sky

Do we still need musicians? Or composers? Why do we need an audience? Since the 1950's, new music has alienated itself completely from society and now only operates within an almost immeasurably small niche of enthusiasts. The general public has no knowledge or love of it. At least, that is what has been alleged for decades on a mainly cultural-political level. Even if this is true, it applies mainly to the traditional concert music presented in concert halls. Electronic music originally formed a small part of that, but on closer inspection has developed much more broadly and is increasingly present in all sorts of layers of society, from musical contexts to social sectors, from concert halls to healthcare, and from the public domain to computer games.

Society is anyway constantly on the move, and connections are made more and more frequently between science and art in solving social issues. Art – and therefore also music – is used increasingly often outside the context of the regular art scene. A shining example is the ever-growing reuse of old buildings, factories and company premises as creative breeding grounds with a clear role in urban development. They form transitional areas, which breathe new life into those older parts of the city through a more informal art practice with more direct public participation, and thus broader social relevance.

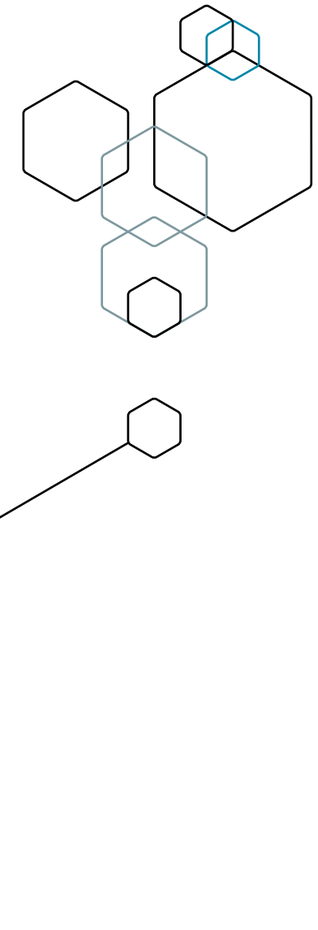
The practice of (art) music is becoming increasingly multidisciplinary. Composers are making more use of a mix of instruments, electronics and video, etc., and concerts are becoming more of an experience or event, just like the more accessible electro scene. The greater flexibility in presentation venues and the link to other arts and contexts is also leading to a different relationship with the audience. These developments thus demand a new attitude, new competencies and new skills from composers and musicians.

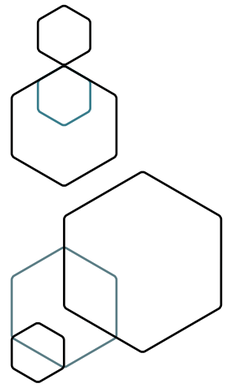
Is this steadily narrowing the gap between research, art and society? Is art music becoming more of a community art as well? Is there a continuum between accessible electronic music and electronic art music, and if so how do they relate to one another? Or is the situation outlined above a social development, alongside which a separate form of autonomous art music can continue to exist as a niche?

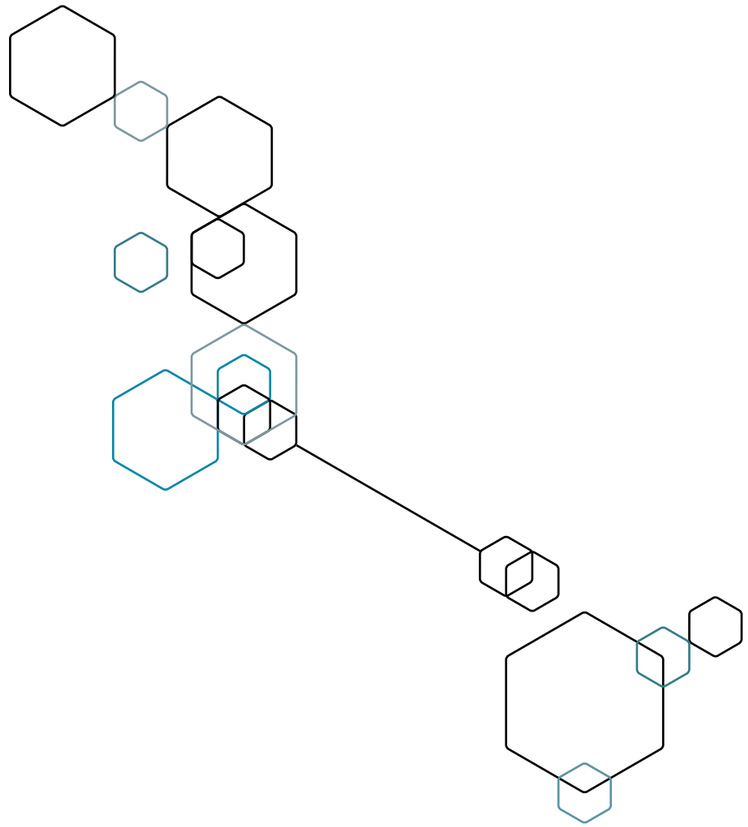


In this development, will the composer become more of a 'maker'? A co-creator? A 'designer'? A mediator? A researcher? And is the research component of electronics and electronic art even more relevant, as it can be used in a broader social context?

Index







Papers

ICMC 2016

InMuSIC: an Interactive Multimodal System for Electroacoustic Improvisation

Giacomo Lepri
STEIM - Institute of Sonology, Royal Conservatoire in The Hague, The Netherlands
leprotto.giacomo@gmail.com

ABSTRACT

InMuSIC is an Interactive Musical System (IMS) designed for electroacoustic improvisation (clarinet and live electronics). The system relies on a set of musical interactions based on the multimodal analysis of the instrumentalist's behaviour: observation of embodied motion qualities (upper-body motion tracking) and sonic parameters (audio features analysis). Expressive cues are computed at various levels of abstraction by comparing the multimodal data. The analysed musical information organises and shapes the sonic output of the system influencing various decision-making processes. The procedures outlined for the real-time organisation of the electroacoustic materials intend to facilitate the shared development of both long-term musical structures and immediate sonic interactions. The aim is to investigate compositional and performative strategies for the establishment of a musical collaboration between the improviser and the system.

1. INTRODUCTION

The design of IMS for real-time improvisation poses significant research questions related to human computer interaction (e.g. [1]), music cognition (e.g. [2]), social and cultural studies (e.g. [3]). An early important work is George Lewis' *Voyager* [4]. In *Voyager*, the author's compositional approach plays a crucial role: specific cultural and aesthetic notions are reflected in the sonic interactions developed by the system. More recently, systems able to generate improvisations in the style of a particular performer (e.g. Pachet's *Continuator* [5] and *OMax* from IRCAM [6]) were developed. In these systems, the implementation of a particular type of finite-state machine, highly refined for the modelling of cognitive processes, allows for the simulation of humanised behaviours such as imitation, learning, memory and anticipation.
In this field of research, the chosen framework for the composition of sonic interactions reflects particular cultural and musical models, performative intuitions, as well as specific cognitive paradigms and technological notions. Music improvisation is here conceived as a wide-ranging creative practice: a synthesis of intricate processes involving physicality, movement, cognition, emotions and sound. The design approach of InMuSIC derived from an embodied cognition of music practice [7]. The majority of the interactive system for improvisation developed during the last years are not based on an embodied cognition of music practice and they focus on the sonic aspects of the performance. Nevertheless, a multimodal approach for the design of improvising IMS was adopted within various research. For example, Ciufo [8], Kapur [9] and Spasov [10] developed IMS able to extract in real-time both gestural and sonic qualities of the performer interacting with the machine. However, these applications are concerned with the recognition of specific body parts and particular gestures (e.g. hands movements). One of the main goal of the presented research is related to the definition of strategies for a qualitative analysis of upper-body features pertinent to a wide range of gestures, not restricted to specific types of movement. This paper presents the system's overall design approach sketching a strategy for the real-time multimodal analysis and representation of instrumental music practice.

2. THE INTERACTIVE FRAMEWORK

The notion of interaction here investigated is inspired by the spontaneous and dialogical interactions characterising human improvisation. The intention is to provide the system with an autonomous nature, inspired by the human ability to focus, act and react differently in relation to diverse musical conditions. In regards to each specific performance, the close collaboration between the musician and InMuSIC should enable the constitution and emergence of specific musical forms. The generation, modification and temporal organisation of new sonic materials are established negotiating the musical behaviour of the performer and the systems internal procedures. In order to facilitate the development of a spontaneous musical act, the platform should then be able to assess different degrees of musical adaptiveness (e.g. imitation/variation) and independence (e.g. contrast/discontinuity). InMusic has been conceived for real-time concert use within contexts related to electroacoustic improvisation. The compositional research has developed alongside a specific musical aesthetic concerned with the exploration of sonic spectral qualities within flexible fluctuations in time rather than actual melodic/harmonic progressions and metrical tempo [11].
The IMS presented relies on the analysis and comparison of sonic and motion qualities. This is by identifying and processing abstracted expressive musical hints of the performer. The attempt of composing and exploring sonic and gestural interdependences is the foundation of the inquired interactive paradigm. Thus, the framework composed to frame and shape the musical interactions, in addition to the sonic dimension, aims to take into account fundamental performative and expressive aspects complementary to the sound production.

3. THE COMPOSITIONAL MODEL

In this section, the InMuSIC's conceptual model is presented. Figure 1 illustrates a layered model based on the work of Leman and Camurri [12]. It is composed of five modules located on three different levels of abstraction, ranging from the representation of physical energy to the more compositional extent related to performative intuitions. Consequently, it is possible to conceive a *continuum* linking the physical world to its musical interpretation. The lowest level is associated to those units that perform tasks related to the physical domain (i.e. detection of sound and movements). The highest level is related to the more abstract components of the system, responsible for compositional choices that govern the real-time sonic interactions. This representation defines an interactive loop and it offers the possibility to frame the essential functions associated to the musical behaviour of the system.
In addition, the conceptual model presented is inspired by the work of von Bertalanffy [13]. The design approach of the relations between the various system's units is influenced by specific criteria: (i) any change in a single unit causes a change in all the units, (ii) the system's behaviour reacts to the incoming data and modifies them in order to either cause change, or to maintain the stationary state (positive and negative feedback) and (iii) the same results may have different origins (i.e. the same causes do not produce the same effects, and *vice versa*). The individual modules will be now briefly introduced.

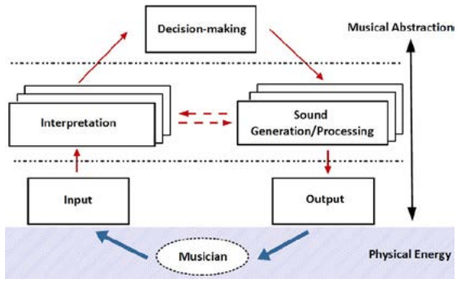


Figure 1. The conceptual model of InMuSIC.

- **Input** - The module executes two main functions: (i) detection of the movements and sounds articulated by the musician and (ii) conversion of this energy (i.e. kinetic and sonic) into digital information. InMuSIC foresees the use of two sensors: the instrument's sound is detected using a condenser microphone and the movement of the performer is captured using the 3D sensor Microsoft Kinect 2.

- **Interpretation** - The information is here interpreted through several parallel processes. Specific sonic and movement features are derived. The comparison of the various analyses provides a second level of interpretation related to the musician's behaviour. In particular conditions, the unit analyses the interventions generated by the system itself. This feedback contributes to the system's self-organisation processes.

- **Decision-making** - The module is located on the highest level of abstraction within the model. Its main function concerns the time-based organisation of the procedures for the generation and manipulation of new sound materials. The decision-making strategies are based on a *negotiation* between the system's internal stochastic processes and the analysed performer's behaviour.

- **Sound generation/processing** - The unit consists of a set of algorithms for sound synthesis and processing: the electronic materials proposed by the system are here actually generated and shaped. In order to establish direct interactions, the system can assign the control of the parameters of the algorithms directly to the data extracted from the modules related to the sound and movement analyses.

- **Output** - The module transfers into the physical domain the information generated by the most abstract units. The processes involved are: (i) the amplification of the generated signal, (ii) the signal's conversion from digital to analogue and (iii) the projection of the sound in the performative space.

4. THE SYSTEM ARCHITECTURE

From a practical point of view, whilst a musician plays a freely improvised session, the system performs five main tasks: movement analysis, sound analysis, sound and movement comparison, decision-making and sound generation. Specific software units compute each of these tasks. The various components are implemented using Max/MSP and EyesWeb. The two platforms communicate through an Open Sound Control (OSC) protocol. A description of the five modules and their functions will now be presented.

4.1 Sound analysis

The unit extracts three low-level audio features: loudness, onset detection and fundamental frequency. The audio signal is analysed by matching and evaluating the outputs of several algorithms [14, 15, 16]. Each of these is tuned for specific dynamic and frequency ranges.
A first level of analysis is associated to the variation in time of the detected data. Initially the features are interpreted through different low-pass filtering and moving average processes. Subsequently the derivative of each feature is computed. By mapping the obtained values using different logistic functions, two thresholds are fixed. In relation to the data previously analysed, the information extracted is defined by three possible states: higher, lower or stable. Consequently, this procedure displays a minimal representation of each audio feature: (i) high, low or stable dynamics (*crescendo vs. diminuendo*); (ii) high, low or stable onset detection (increase *vs.* decrease of the events density); (iii) high, low or stable pitch deviation (expansion *vs.* reduction of the used frequency range). The algorithms implemented interpret the incoming values by means of an *inertial behaviour*. In order to detect any positive or negative change, a certain amount of variation is required. This conduct, simulating the function of a short-term memory, is specifically calibrated for each feature. This is crucial to the fine-tuning of the system's sensitivity.

The understanding of the performer's sonic behaviour is therefore associated to the variation in time of the extracted features. The methodology adopted is influenced by psychological research on human communication [17]. The main assumption is that we can only perceive the relationships or models of relationships that substantiate our own experience. Our perceptions are affected by processes of variation, change or motion. Any phenomenon is perceived only in relation to a reference: in this case the music previously played.

4.2 Movement analysis

Based on the research by Glowinski et al. [18] for the analysis of affective nonverbal behaviour using a reduced amount of visual information, the module extracts expressive gestural features. This interpretation implies the analysis of behavioural features pertinent to a wide range of gestures and not restricted to specific types of movement. The challenge consists of detecting information representative of an open sphere of possible expressive motions: the chosen strategy focuses on a minimal representation of affective movements. A qualitative approach to the analysis of upper-body movements and affect recognition, is hereby adopted [19]. Considering a reduced amount of visual information (i.e. 3D position, velocity, and acceleration of the musicians head, hands and elbows - see 2), three expressive features are extracted: smoothness (degree fluidity associated to the head movement), contraction index (degree of posture openness) and quantity of motion (QOM) (overall kinetic energy).
Applying the same procedure, illustrated in the sound analysis section, the features are further interpreted. Each analysis is reduced to three possible states: (i) high, low or stable smoothness (detection of fluidity and continuity *vs.* jerky or stillness in regards to the head movements); (ii) high, low or stable QOM (overall QOM variation - presence of motion *vs.* stillness or isolated movements); (iii) high, low or stable contraction index (variations in the degree of posture - open *vs.* close).

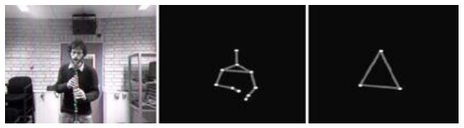


Figure 2. The detected skeleton of a musician playing the clarinet. The motion analysis is based on a minimal representation of affective gestures.

4.3 Sound and movement comparison

The module is designed to combine and compare the data coming from the movement and sound analyses. The various *stable* states are ignored: the detection of a *stable* state does not produce any change to the internal conditions of the system (i.e. maintenance of the current stationary state). Figure 3 illustrates the available combinations in regard to each *high-low* state. Through a Graphical User Interface (GUI) it is possible to manually select which combinations the module will consider during the performance. Figure 3 presents a possible selection of the states combinations often used by the author performing with InMuSIC. Once a specific combination is chosen (e.g. low QOM and low loudness), the unit constantly verifies if the two states are simultaneously detected: to each selected combination, a simple boolean condition is applied. In addition, the unit tracks how long each condition is verified. In short, during the performance, the data sent to the decision-making module defines (i) which condition selected is currently true and (ii) the time associated to the persistence of each verified condition.
The computation of the various *high-low* states allows for the gathering of information related to the variation in time of the extracted features (continuous inertial interpretation). For instance, in regards to the past trends, the QOM is now increasing or decreasing. The combination and comparison of the *high-low* states associated to the various features is conceived as a further level of abstraction within the expressive analysis of the performer. The organisation of the processes for the generation of new electronics interventions is therefore related to the detection of specific *high-low* conditions (finite-state machine like behaviour). The strategy implemented aims to achieve a minimal and qualitative interpretation of instrumental music practice: the focus is oriented to analyse *how* the musician plays instead of *what* the musician plays.

		Loudness		Events Density		Pitch Deviation	
		high	low	high	low	high	low
QOM	low		✓				
	high			✓			
Contraction Index	low			✓			
	high	✓					✓
Smoothness	low						
	high		✓				✓

Figure 3. The possible comparisons of sound and movement analyses. The ticked boxes are the combination often used by the author while performing with the system.

4.4 Decision-making

The function of the unit mainly concerns the time-based organisation of new musical information (e.g. activation, duration, cross fade and muting of the various system's *voices*). Here the main focus is oriented towards the composition of decision-making processes allowing for the development of both long-term musical structures and immediate sound interventions. The unit establishes sonic interactions that develops inside a *continuum* ranging between two different temporal durations: from short-term immediate re-actions (maximum duration of 4 seconds), to long-term re-actions (maximum duration of 4 minutes). The reference paradigm refers to studies on human auditory memory [20] (short-term and long-term). An *awareness* of different *real-times* is here sought. The overall timing of the unit (i.e. the actual clock that triggers the various sonic processes) is controlled by an irregular *tactus* generated by a stochastic process. The rate of this clock is constantly modified by the variation in time of the onset analysis: the system's *heart beat* increases when the performer articulates a music dense of sonic events and *vice versa*.
The generation and organisation of both short-term and long-term interventions is associated to the detection of the *high-low* conditions occurring during the performance (e.g. simultaneous detection of low QOM and low loudness). To each condition a set of sound processes is applied, a particular type of synthesis can be associated to

more then one condition. The more a condition is detected, the higher the probability is to trigger the related sound processes. Furthermore, stochastic procedures influence the relative weight of each probability with a specific set. The duration of an active sonic process is affected by the persistence in time of the associated *high-low* condition.
Simultaneously, the unit regulates two further parallel procedures. Once a particular sound process is activated, timbral adjustments can occur. The unit can establish a direct link between the performers sonic and gestural behaviours and the processes for the sound synthesis. This relates to the modification of current electronic materials (i.e. manipulation of the control-rate data associated to the triggered sound) using the information coming from the sound and movement analyses. During the performance, the unit can also send the produced electronic materials to the sound analysis module. Thus, a feedback process is activated: instead of evaluating the sonorities produced by the musician, InMuSIC analyses its own output. This situation mainly takes place when the performer is not playing. The possibility of 'listening to itself' is conceived as a further degree of autonomy within the system's agencies.
The described procedures enables the potential generation of a wide range of *musical narratives*, emerging and evolving with regards to each specific performance.

4.5 Sound generation

The sound generation module is conceived to produce heterogeneous sound materials. The sonic interactions generated entail a multiplicity of possible changes concerning diverse musical circumstances. In relation to the different performative and expressive contexts, the variety of timbral and sonic articulation appears to be an important requirement for the development of an engaging interactions. The algorithms implemented for the generation of the electronic materials can be organised into three categories: (i) synthesis (FM, additive, subtractive and physical models [21]), (ii) sampling (real-time processing of pre-recorded sounds) and (iii) live processing (live sampling, live granulation, Fast Fourier transform analysis and re-synthesis and reverberation).
The individual techniques used can be conceived as system's *voices*. Each *voice* is characterised by specific qualities, that are spectro-morphological (i.e. related to the distribution of energy inside the sonic spectrum) and gestural (i.e. associated to the articulation and transformation of sound material over time). In relation to the generated sonorities, each algorithm has been designed to guarantee a certain degree of indeterminacy. The goal is to define processes able to develop extensive variations and manipulations of the electronic materials within predefined physical scopes (e.g. frequency, dynamic and temporal ranges). In other words, every single voice is conceived to explore diverse *sound spaces*. The musician is invited to *navigate* these timbre spaces [22] in collaboration with the system. Once a voice is active, timbre variations may occur: these changes are shaped by the external interventions inferred by the performer's musical behaviour. The intention is to develop a close dialogue/collaboration between acoustic and electronic materials (e.g. fusion, separation, imitation, variation and contrast). This approach allows to partially solve a dichotomy that emerges when attempting to combine the practices of composition and improvisation. Through the real-time interactions with the performer, InMuSIC organises and shapes pre-composed musical materials. The challenge relies on balancing the processes that leads to the development of musical forms within a *performative time* and the musical choices previously made over a *compositional time*.

5. THE PERFORMANCE

InMuSIC has been extensively used by the author in live concerts and it has been presented in several musical events and research contexts.The performance was often evaluated as engaging and successful. The sonic variety generated and the system responsiveness appear to be the most valued traits of the IMS here presented.
InMuSIC was also tested by five expert improvisers in informal settings. The aim was to explore the use of InMuSIC with different players and instruments (two clarinettists, one trombonist, one cellist and one pianist). After a short introduction, the musicians were invited to freely play with the system. Open interviews were undertaken to investigate their impressions. The system was essentially perceived as a generative algorithm allowing for a shared exploration of interesting and engaging musical materials. The experience of playing with InMuSIC was compared to a conversation with a little child: "You don't know very well how it will react. Its a little bit shy at first and you have to draw something out of it". The system was also perceived as able to play both in foreground (leading) and background (either following or leaving space for solos), although some musician felt that InMuSIC was leading too often. Some improvisers perceived a not always bidirectional interaction: the machine was "not listening much". Furthermore, they expressed the desire for a IMS that would more frequently retrieve and develop the materials proposed by them.
Some musicians were slightly frustrated by the impossibility of clearly understand and control the functioning of InMuSIC. Others referred to this aspect positively comparing this situation to the real human-human interaction. Interestingly, some musicians observed that, during the performance, a turning point occurred. After a first clear and simple interaction (i.e. direct action-reaction relationship) the musicians changed their attitude. Once recognised that the machine was listening and responding (even if not constantly) they started to better engage with the system being more open to the electronic material proposed.
During the sessions, the algorithms for the sound and movement analysis were not modified: the settings normally used by the author performing with the clarinet were kept. Compared to the author experience with InMuSIC, it was noticed that the system was less reactive and always performing with a reduced amount of sonic possibilities. This might suggest that the system has to be tuned according to each specific player. In addition, all the musicians agreed on the need of rehearsing in order to achieve a more satisfying performance. There were no significant differences in the system outcome while playing with different instruments. This might be related to the qualitative approach adopted for the analysis of musical behaviour (i.e. looking at how do we play instead of what do we play).

6. CONCLUSIONS

InMuSIC is a multimodal interactive system for electroacoustic improvisation (clarinet and live electronics). It can be defined as a system that composes/improvises music through a dialogical modality. The aim of the research is to design a platform able to establish a close collaboration with the performer, in relation to the analysed musical information. Music improvisation is here conceived as a spontaneous expressive act involving cognitive and technical skills conveyed by sonic and physical behaviours. The interactive paradigm developed is therefore based on the combination and comparison of the performers movement and sound analyses. InMuSIC is tuned to be sensitive to a specific *apparatus* of gestural and sonic behaviours, according to both the instrumental practice of the clarinet and the performative attitudes characterising the author's expressiveness. Future developments of the system may include the possibility of expanding this *apparatus* in order to explore diverse audio and gestural features and widen the performer's analysis. It is not the intention of the author to categorise or attribute any specific semantics to the various expressive cues represented. Instead, the interest relies on the exploration and use (or abuse) of these musical indications in the contexts of composition and improvisation. Nevertheless, the author's impression is that, with a more systematic approach, the multimodal analysis presented might allow for the revealing of performative traits pertinent to specific instruments and players. The conceived performance presumes the development of both musical structures and immediate re-action, emerging from the human-computer cooperation.

7. REFERENCES

[1] A. Cont, S. Dubnov, and G. Assayag, "A framework for anticipatory machine improvisation and style imitation," in *Anticipatory Behavior in Adaptive Learning Systems (ABiALS)*. ABIALS, 2006.

[2] A. R. Addessi, "From Econ to the mirror neurons: Founding a systematic perspective on the reflexive interaction paradigm," *ICMPC-ESCOM2012 Proceedings*, pp. 23–28, 2012.

[3] G. E. Lewis, "Interacting with latter-day musical automata," *Contemporary Music Review*, vol. 18, no. 3, pp. 99–112, 1999.

[4] ——, "Too many notes: Computers, complexity and culture in voyager," *Leonardo Music Journal*, vol. 10, pp. 33–39, 2000.

[5] F. Pachet, "The continuator: Musical interaction with style," *Journal of New Music Research*, vol. 32, no. 3, pp. 333–341, 2003.

[6] G. Assayag, G. Bloch, M. Chemillier, A. Cont, and S. Dubnov, "Omax brothers: a dynamic yopology of agents for improvization learning," in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*. ACM, 2006, pp. 125–132.

[7] M. Leman, *Embodied music cognition and mediation technology*. Mit Press, 2008.

[8] T. Ciufo, "Design concepts and control strategies for interactive improvisational music systems," in *Proceedings of the MAXIS International Festival/Symposium of Sound and Experimental Music*, 2003.

[9] A. Kapur, "Multimodal Techniques for Human/Robot Interaction," in *Musical Robots and Interactive Multimodal Systems*. Springer, 2011, pp. 215–232.

[10] M. Spasov, "Music Composition as an Act of Cognition: ENACTIV–interactive multi-modal composing system," *Organised Sound*, vol. 16, no. 01, pp. 69–86, 2011.

[11] D. Smalley, "Spectromorphology: explaining soundshapes," *Organised sound*, vol. 2, no. 02, pp. 107–126, 1997.

[12] M. Leman and A. Camurri, "Understanding musical expressiveness using interactive multimedia platforms," *Musicae Scientiae*, vol. 10, no. 1 suppl, pp. 209–233, 2006.

[13] L. V. Bertalanffy, "General system theory: Foundations, development, applications," Braziller. New York, Tech. Rep., 1968.

[14] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[15] M. Malt and E. Jourdan, "Real-Time Uses of Low Level Sound Descriptors as Event Detection Functions Using the Max/MSP Zsa. Descriptors Library," *Proceedings of the 12th Brazilian Smposium on Computer Music*, 2009.

[16] T. Jehan and B. Schoner, "An audio-driven perceptually meaningful timbre synthesizer," *Analysis*, vol. 2, no. 3, p. 4, 2002.

[17] P. Watzlawick, J. B. Bavelas, D. D. Jackson, and B. O'Hanlon, *Pragmatics of human communication: A study of interactional patterns, pathologies and paradoxes*. WW Norton & Company, 2011.

[18] D. Glowinski, N. Dael, A. Camurri, G. Volpe, M. Mortillaro, and K. Scherer, "Toward a minimal representation of affective gestures," *Affective Computing, IEEE Transactions on*, vol. 2, no. 2, pp. 106–118, 2011.

[19] A. Camurri, I. Lagerlöf, and G. Volpe, "Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques," *International journal of human-computer studies*, vol. 59, no. 1, pp. 213–225, 2003.

[20] B. Snyder, *Music and memory: An introduction*. MIT press, 2000.

[21] D. Trueman and R. DuBois, "PeRColate: A Collection of Synthesis," *Signal Processing, and Video Objects for MAX/MSP/Nato*, vol. 1, p. b3, 2009.

[22] D. L. Wessel, "Timbre space as a musical control structure," *Computer music journal*, pp. 45–52, 1979.

Molecular Sonification of Nuclear Magnetic Resonance Data as a Novel Tool for Sound Creation

Falk Morawitz
University of Manchester
falk.morawitz@postgraduate.manchester.ac.uk

ABSTRACT

The term molecular sonification encompasses all procedures that turn data derived from chemical systems into sound. Nuclear magnetic resonance (NMR) data of the nuclei hydrogen-1 and carbon-13 are particularly well suited data sources for molecular sonification. Even though their resonant frequencies are typically in the MHz region, the range of these resonant frequencies span only a few tens of kHz. During NMR experiments, these signals are routinely mixed down into the audible frequency range, rendering the need for any additional frequency transpositions unnecessary. The structure of the molecule being analysed is directly related to the features present in its NMR spectra. It is therefore possible to select molecules according to their structural features, in order to create sounds in preferred frequency ranges and with desired frequency content and density. Using the sonification methodology presented in this paper, it was possible to create an acousmatic music composition based exclusively on publicly accessible NMR data. It is argued that NMR sonification, as a sound creation methodology based on scientific data, has the potential to be a potent tool to effectively contextualize extra-musical ideas such as Alzheimer's disease or global warming in future works of art and music.

1. MOLECULAR SONIFICATION IN ART, SCIENCE AND MUSIC

In its widest sense, the term molecular sonification includes all procedures that turn data derived from chemical systems into sound. These chemical systems may be single atoms, small molecules, or macromolecules such as proteins or DNA. Although it is possible to sonify some atomic properties in real time, most of the sonification methodology involves turning pre-recorded spectra, or spectral information, into sound.

Scientifically, molecular sonification has been used to analyse large DNA datasets [1], or to find visually imperceptible changes in coupled atomic oscillations [2]. Generally, however, its use in a scientific context is extremely sparse. Contrastingly, molecular sonification has been utilized in various multi-media installations as well as in purely instrumental and acousmatic compositions. In many of those works, molecular systems were sonified 'indirectly', for example by assigning musical tones and rhythms to DNA sequence combinations [3], or by assigning musical parameters to size, velocity and positions of atomic clusters [4].

However, molecular systems can be sonified 'directly', too, by turning atomic resonant processes measured in analytical chemical experiments directly into sound. Despite a plethora of different spectroscopic methods being available, sources used for direct sonification to date have been almost entirely limited to infra-red spectra. Infra-red spectroscopy measures the vibrational behaviour of atoms and molecules and it has been investigated for its use as a sound source for molecular sonification in theoretical and in applied musical contexts. [5, 6, 7].

One common feature of all these sonification procedures is that artistic choices have to be made during the sonification process: notes or pitches have to be assigned to different chemical features, or choices have to be made on how to transpose infra-red frequencies, typically of many trillions of hertz, into the audible spectrum.

In this paper, nuclear magnetic resonance (NMR) spectroscopy, a standard analytical method in organic chemistry, is presented as a novel and unexplored data source for molecular sonification. In contrast to infra-red spectroscopy, in modern NMR experiments the frequencies of the nuclear signals are converted directly into the audible range during the recording process, enabling a direct translation of data into sound, sometimes described as audification [8]. Here, the key physical principles of NMR spectroscopy are introduced. The spectral characteristics of hydrogen-1, denoted ^{1}H, and carbon-13, ^{13}C, NMR spectra are described, and different sonification strategies are discussed. The use of sounds made via sonification of NMR data in acousmatic music composition is contextualized.

2. A MUSICIAN'S INTRODUCTION TO NMR SPECTROSCOPY

Nuclear magnetic resonance is mostly utilized in structure elucidation and validation, as NMR measurements are highly sensitive to structural changes in molecules. For example, Figures 1 and 2, on the next page, show the ^{1}H NMR spectra of diethyl ether and methyl n-propyl ether, with the structure formulae CH_3-CH_2-O-CH_2-CH_3 and CH_3-O-CH_2-CH_2-CH_3, respectively. Even though the two ethers have very similar structures and contain the same numbers and types of atoms, their NMR spectra are very different.

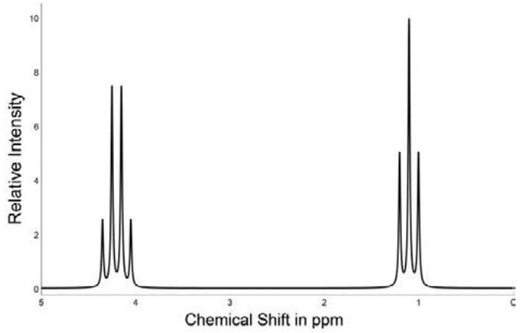


Figure 1. Simulated ^{1}H NMR spectrum of diethyl ether, CH_3-CH_2-O-CH_2-CH_3. Note: Conventionally, NMR spectra are drawn with increasing frequency from right to left. In this case 1 ppm equals 500 Hz.

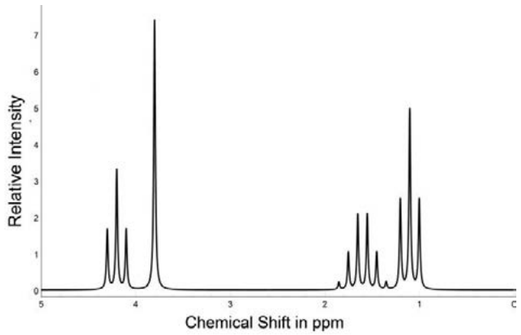


Figure 2. Simulated ^{1}H NMR spectrum of methyl propyl ether, CH_3-O-CH_2-CH_2-CH_3. Note: 1 ppm = 500 Hz.

It is impossible to accurately explain the mechanisms and principles behind NMR analysis without introducing a vast amount of scientific terms and concepts such as ground state nuclear spin, Larmor frequency or transverse magnetisation. A thorough scientific explanation is beyond the scope of this paper and it is recommended for interested readers to consult M. Levitt's excellent book 'Spin Dynamics' [9]. In simple terms, for a given element with a magnetic nucleus, each atom in a molecule has its own resonance, which is split into a set of resonances with slightly different frequencies if there are other magnetic nuclei nearby. The resonances are measured by placing a sample in a strong magnetic field, then applying a short powerful pulse of radiofrequency and recording the "ringing" of the nuclear spins. Essentially, we can compare the molecules in the sample with tiny bells that are made audible by being hit with a radio frequency hammer.

The signal that is measured is known as free induction decay, or FID, which is subsequently Fourier transformed to yield the NMR spectrum of the sample, as seen in Figure 3. It is possible to convert either the FID or the NMR spectrum into sound, as explained in section 4: "sonification methodology".

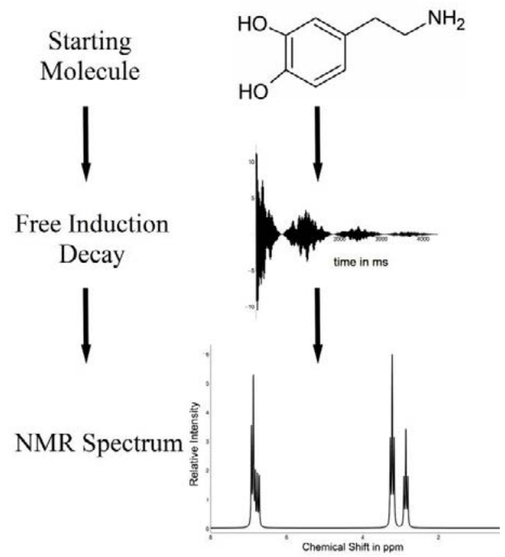


Figure 3. The chemical dopamine and its corresponding ^{1}H FID and ^{1}H NMR spectrum.

While NMR spectroscopy can detect any atom with an odd number of neutrons or protons, two isotopes especially interesting for sonification are ^{1}H (the proton) and ^{13}C. These nuclei are by far the most commonly used in organic chemistry, with many hundred thousands of datasets available online. The Human Metabolome Database alone hosts spectra for more than 40,000 different chemicals found in the human body [10].

3. CHARACTERISTICS OF SONIFIED NUCLEAR MAGNETIC RESONANCE SPECTRA

3.1 ^{1}H NMR

Figures 1 and 2 show typical shapes of a ^{1}H NMR spectrum. In order to convert the values normally displayed in units of ppm (deviation from a reference in parts per million) to Hertz, we have to know the reference oscillating frequency of a proton, which in turn is dependent on the magnetic field strength of the NMR machine used and the reference frequency chosen. Here we will assume that the reference frequency is set to zero ppm. In modern NMR experiments the magnetic field strength will very likely correspond to an oscillation frequency of either 500 or 600 MHz, which means that a 1 ppm difference in chemical shift will be equal to 500 or 600 Hertz, respectively. Knowing this conversion, it can be seen that virtually all ^{1}H NMR peaks are situated within the range of 0 – 6000 Hz with most peaks typically lying in between 600 and 4000 Hz.

Depending on which atoms and structures are present in a molecule, frequency clusters will occupy distinct frequency ranges. For example, proton signals associated

with carbohydrates will normally exhibit frequencies around 1 – 3 ppm (500 – 1500 Hz) while unsaturated and aromatic hydrogens, or hydrogens connected to very electronegative atoms, are shifted upwards to 5 – 8 pm (2500 – 4000 Hz). Figure 4 shows the ^{1}H NMR spectrum of ethyl benzene, a molecule with both low and high frequency content, and the assignment of its hydrogens to their corresponding frequency clusters.

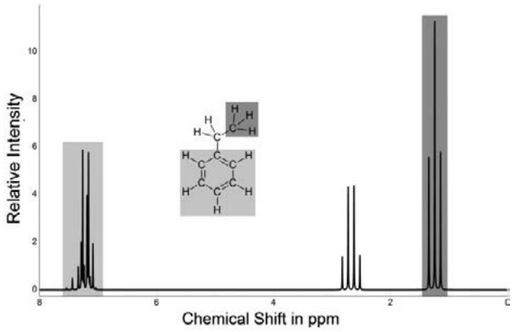


Figure 4. Ethyl benzene contains high frequency hydrogens attached to the aromatic benzene ring (lighter grey highlighting), as well as low frequency hydrogens associated with the ethyl group (dark grey highlighting and group not highlighted).

The number of signals per spectrum will depend on the complexity and structure of the molecule, and can be as few as one signal or 1000 or more peaks for very complex molecules. Frequency peaks can be spread out over a wide frequency area, as in Figure 4, or concentrated in narrow regions as seen in Figure 5.

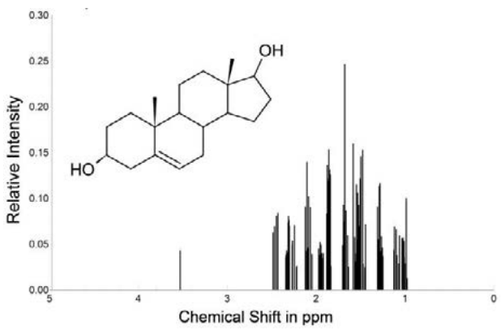


Figure 5. The majority of hydrogens of the molecule dehydroepiandrosterone are connected to saturated carbon atoms, resulting in an accumulation of more than 80 frequency peaks in the narrow range of 1 – 2.5 ppm (500 – 1250 Hz).

^{1}H nuclei are coupled to neighbouring ^{1}H nuclei, causing peaks to be split into "multiplets" by what is known as J - coupling. A single resonance is split into a set of slightly different frequencies, up to a few tens of hertz across. In the simplest cases, resonances are split into $N + 1$ equally spaced signals, whose intensities are given by Pascal's Triangle: the components of a doublet signal have a ratio of roughly 1:1, a triplet peak 1:2:1, a quartet 1:3:3:1, and so on. Different multiplets can be seen in Figure 4: from left to right, a complex multiplet, a quartet and a triplet.

Interestingly, these very closely spaced J-coupled frequencies, a trademark of ^{1}H NMR signals, can lead to strong inherent tremolo-type features in the sound wave due to interference, as seen in Figure 6.

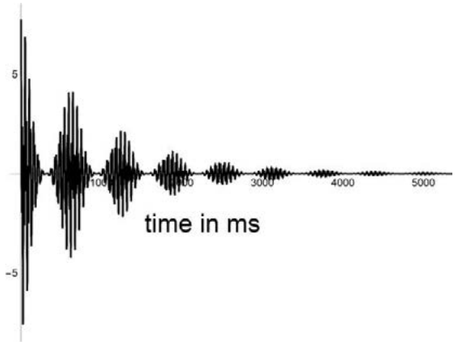


Figure 6. The ^{1}H FID of diethyl ether, showing a strong pulsing.

3.2 ^{13}C NMR

In the most commonly used ^{13}C NMR experiment setting, a deviation of 1 ppm corresponds to a frequency shift of 125 Hz. The frequency range of ^{13}C NMR peaks can be as wide as 0 – 30000 Hz, with most peaks typically lying above 12500 Hz.

Contrary to ^{1}H NMR spectra, the vast majority of ^{13}C spectra are decoupled through the way they are recorded. This means that ^{13}C NMR peak are not split, each appearing as a single frequency. The resulting spectra are an assortment of single sine and cosine waves, with aromatic compounds having a higher frequency content (+12500 Hz) whereas saturated carbohydrate peaks tend to be at lower frequencies (0 – 50 ppm, 0 – 6250 Hz). Figure 7 shows the ^{13}C NMR spectrum of ethyl benzene, a molecule with both low and higher frequency content.

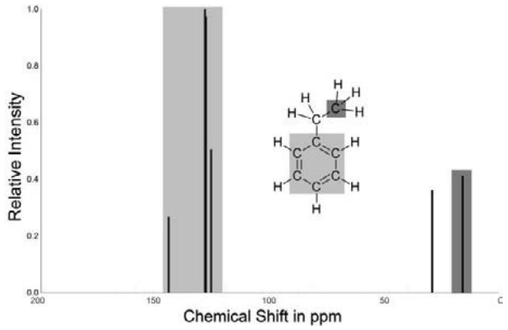


Figure 7. Ethyl benzene contains high frequency carbons (lighter grey highlighting), as well as low frequency carbons associated with the ethyl group (dark grey highlighting and signal not highlighted).

The maximum number of peaks in a carbon NMR spectrum is equal to the number of carbon atoms present in the molecule, with molecules showing symmetry having fewer signals and can range from one to 200 signals or more.

4. SONIFICATION METHODOLOGY

To turn NMR spectra into sound there are two major methods available: it is possible to turn experimental raw data directly into sound (*in situ / in vitro audification*), or to 'reverse-engineer' the sound, from the Fourier transformed and analysed data via additive synthesis, as seen in Figure 8.

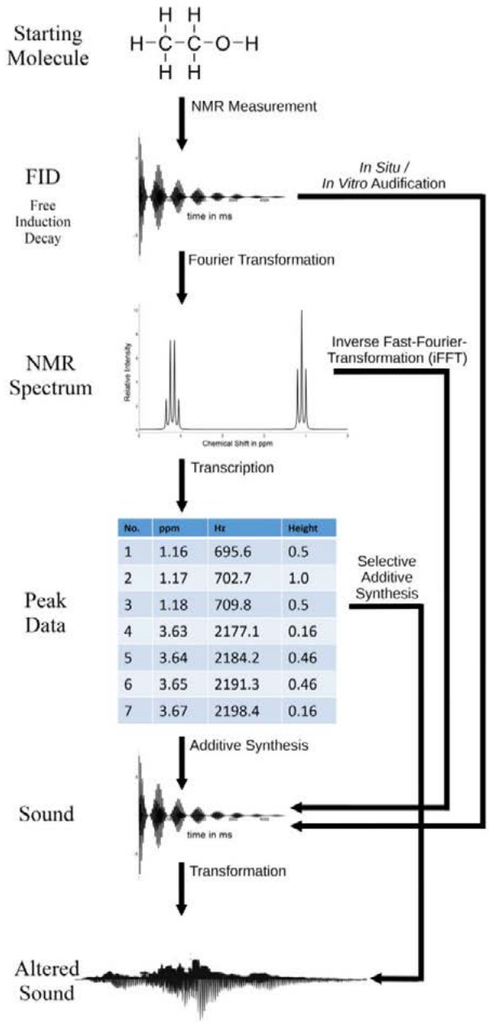


Figure 8. Possible pathways to create authentic and altered sounds from raw and processed NMR data.

4.1 FID audification

The FID produced in an NMR experiment can be audified directly, by direct recording from the output of the spectrometer receiver, by importing recorded FID files into software packages such as the DOSY Toolbox [11], or via custom coding of an audification routine in programs such as Matlab or Mathematica.

By starting with a direct recording of raw experimental data, it can be argued that the sonification of FID data will lead to the most authentic molecular sounds. Sounds made from FID data, however, often contain unwanted additional frequency peaks, arising from small sample impurities or the solvent of the sample itself. Sounds made from FIDs also contain more random background noise than sounds artificially created via additive synthesis, as seen in Figure 9 and 10.

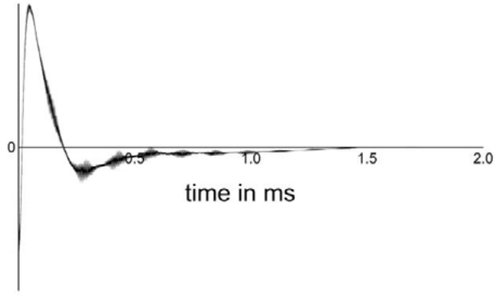


Figure 9. Experimental free induction decay of ethanol, with a strong low frequency oscillation at around 2 Hz present, due to significant water impurities.

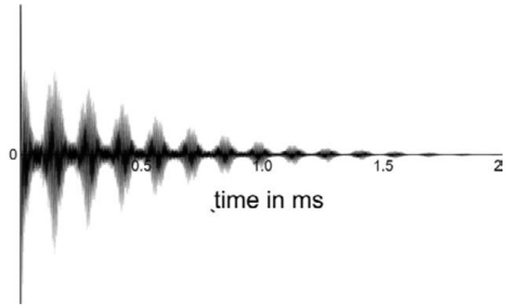


Figure 10. Computer generated free induction decay of ethanol

FIDs generated in a standard ^{1}H NMR measurement often decay within a few seconds, as it can be seen in Figure 9 and 10, limiting their use for long textures and drones without the use of additional sound transformations. A few experimental procedures exist that generate continuous signals or rapidly repeating FIDs. However, data of these specialized experiments are not as readily available online.

4.2 Sonification via additive synthesis

Sonification via additive synthesis will give 'cleaner' results than FID audification, as it is possible to omit unwanted frequencies and experimental noise from the soni-

fication process. Additionally, additive synthesis offers more control over individual sonification parameters. It is possible to selectively sonify chosen frequency clusters, or change the volume envelope and phase for each frequency peak individually. It is possible to sonify different frequency peaks sequentially instead of all at once. This leads to more versatility in the sound creation process and enables the creation of short percussive bursts, bell-type sounds, drones, complex textures and combinations of thereof.

5. MOLECULAR SOUNDS AS THE BASIS FOR MUSIC COMPOSITION

Using molecular sonification to create a collection of fingerprint sounds, their further employment in a creative setting has to lie somewhere in between two extremes: on the one hand, it is possible to leave the raw sounds unchanged. On the other hand, it is feasible to completely change the sonic characteristics of the starting material and shape them into new forms, their origins unrecognizable.

Both approaches have certain advantages and shortcomings: if only raw and unaltered chemical sounds are used, the audience might appreciate the 'scientific authenticity' of the sounds they hear, but could tire quickly of listening to temporally and timbrally similar sounds repeatedly. If the sounds are altered beyond recognition, sophisticated soundscapes can be created, but it would make no difference if the sounds were derived from chemicals or any other system. In that case, why use chemical sounds at all?

To inquire further into the use of molecular sonification in a musical setting, the piece 'Spin Dynamics' was created. 'Spin Dynamics' is an acousmatic piece consisting solely of sounds made via molecular sonification of hydrogen-1 and carbon-13 NMR data. To negotiate between artistic freedom and the scientific authenticity of the sound material, all sounds heard in the composition were transformed subject to the same constraint: at least one characteristic of the original NMR sound had to be left unchanged throughout, while freely changing any other aspect of the sound. For example, the piece begins by introducing a texture with a strong tremolo feature. The texture was made from the raw sound of diethyl ether (seen in Figure 6), keeping the strong tremolo - feature, but completely changing the timbre of the sound by consecutively adding increasing amounts of artificial harmonic and inharmonic partials. In other textures of the composition, the raw molecular timbre was kept unchanged, but swept through with a narrow band pass filter, adding a temporal narrative to an otherwise static texture, only revealing certain frequency clusters at a time.

In general, sounds created from ^{13}C NMR spectra occupied higher frequency bands and were used for the creation of percussion-type sounds. ^{1}H NMR sounds occupy lower frequency bands and are apt for the creation of textures and drones.

6. MOLECULAR SONIFICATION IN THE CONTEXT OF SCIENCE-BASED MUSIC AND ART

Molecular sonification is not just a source of new timbres: many science-based art installations that have been assessed for their public impact have been reported to be considered as 'science and not art' by the audience, if the scientific origin of the work was clear [12]. Cultural expectations of science can strongly shape the impression received by the audience [13], and musical compositions based on scientific data have been described as conveying 'a sort of scientific authority' [14], altering and potentially enhancing the audience's interaction in regards to the aesthetic and meaning of the sound composition. Simon Emmerson notes that by using (mathematical) models, the composer has the means to incorporate non-musical principles into the compositional process. By doing so, the composer 'reanimates' the model and positions them 'in a relationship with us [the audience]' [14].

Molecular sonification, as a sound creation technique based on scientific principles, can therefore be a powerful tool to contextualize extra-musical ideas that are describable through their underlying chemical mechanisms, such as global warming or Alzheimer's disease but only, if the audience can see the scientific origin and authenticity of the work. It can be argued that using experimental NMR data as a source for sonification is more appropriate than any indirect sonification method, or the use of infra-red data, as artistic choices are kept to a minimum during the NMR sonification process, with no need for 'arbitrary' frequency assignments or transpositions.

7. CONCLUSIONS

By carefully selecting the starting chemicals for NMR sonification, one can determine the overall sound aesthetic of the resulting sound, including the amount of high and low frequency content as well as the complexity of the created sound. In general, ^{13}C NMR peaks will occupy higher frequency regions than ^{1}H NMR, and a combination of sounds created using both data sources together will occupy the whole audible spectrum, making a mix of the two very suitable as a basis for the composition of electroacoustic music. The acousmatic piece 'Spin Dynamics' has been created, exploring the aesthetic possibilities of NMR derived sounds.

The use of sounds created through sonification of NMR data in musical compositions and sound art is almost unexplored. To reach its full potential, it is argued that creative works utilizing molecular sonification will need to exhibit a 'scientific rigidity'. How this rigidity can be implemented and conveyed to the audience has not yet been addressed. Future work will explore the use and impact of molecular sonification in different media, from acousmatic compositions to multi-media installations. It will also investigate the use of different mapping methodologies, such as fuzzy logic.

There is a large number of different isotopes that can be used in NMR analysis, for example fluorine-19, phosphorus-31, lithium-7, aluminium-27, hydrogen-2 (deuterium) nitrogen-14, tin-119 or yttrium-89. The use of such isotopes, the use of extended NMR techniques such as 2D, 3D and solid state NMR, as well as the use of molecular sonification in a live context, offer many directions for future research.

Acknowledgments

I would like to express my gratitude to Professor G. A. Morris and Professor R. Climent for their constant support and guidance throughout the development of this work.

8. REFERENCES

[1] K. Hayashi, and N. Munakata, "Basically Musical," *Nature*, vol. 310, pp 96, 1984.

[2] S. V. Pereverze, A. Loshak, S. Backhaus, J.C. Davis, and R.E. Packard, "Quantum oscillations between two weakly coupled reservoirs of superfluid ^{3}He," *Nature*, vol. 388, pp. 143, 1997.

[3] J. Marc, "Composing DNA Music Within the Aesthetics of Chance, Perspectives of New Music," *Perspectives of New Music*, vol. 46, no. 2, pp. 243-259, 2008.

[4] D. Glowacki, "Using Human Energy Shields to Sculpt Real Time Molecular Dynamics," *Molecular Aesthetics*, vol. 1, pp. 246–257, 2013.

[5] T. Delatour, "Molecular Music: The Acoustic Conversion of Molecular Vibrational Spectra," *Computer Music Journal*, vol. 24, no. 3, pp. 48-68, 2000.

[6] T. Delatour, "Molecular Songs," *Molecular Aesthetics*, vol. 1, pp. 293 – 311, 2013.

[7] S. Alexjander, and D. Deamer, "The Infrared Frequency of DNA bases: Science and Art," *IEEE Engineering in Medicine and Biology Magazine*, vol. 18, no. 2, pp. 74-79, 1999.

[8] B. Truax, "Sound, Listening and Place: The aesthetic dilemma," *Organised Sound*, vol. 17, no. 3, pp. 193–201, 2012.

[9] M. H. Levitt, *Spin Dynamics*, Wiley, 2008.

[10] D. S. Wishart, T. Jewison, A. C. Guo, M. Wilson, and C. Knox, "HMDB 3.0 - The Human Metabolome Database in 2013," *Nucleic Acids Res.,* vol. 1, p. 41, 2013.

[11] M. Nilsson, "The DOSY Toolbox: A new tool for processing PFG NMR diffusion data," *Journal of Magnetic Resonance*, vol. 200, pp. 26-302, 2009.

[12] A. Vandso, "Listening to the world," *SoundEffects*, vol. 1, no. 1, pp. 67 – 81, 2011.

[13] L. Meyer, "Emotion and the Meaning in Music," vol. 1, no. 1, p. 43, 1970.

[14] S. Emmerson, *Living Electronic Music*, p. 39, Routledge, 2007.

The Effects of Reverberation Time and Amount on the Emotional Characteristics

Ronald Mo, Bin Wu, Andrew Horner
Department of Computer Science and Engineering,
Hong Kong University of Science and Technology,
Hong Kong
ronmo@cse.ust.hk, bwuaa@cse.ust.hk, horner@cse.ust.hk,

ABSTRACT

Though previous research has shown the effects of reverberation on clarity, spaciousness, and other perceptual aspects of music, it is still largely unknown to what extent reverberation influences the emotional characteristics of musical instrument sounds. This paper investigates the effect the effect of reverberation length and amount by conducting a listening test to compare the effect of reverberation on the emotional characteristics of eight instrument sounds over eight emotional categories. We found that reverberation length and amount had a strongly significant effect on Romantic and Mysterious, and a medium effect on Sad, Scary, and Heroic. Interestingly, for Comic, reverberation length and amount had the opposite effect, that is, anechoic tones were judged most Comic.

1. INTRODUCTION

Previous research has shown that musical instrument sounds have strong and distinctive emotional characteristics [1, 2, 3, 4, 5]. For example, that the trumpet is happier in character than the horn, even in isolated sounds apart from musical context. In light of this, one might wonder what effect reverberation has on the character of music emotion. This leads to a host of follow-up questions: Do all emotional characteristics become stronger with more reverberation? Or, are some emotional characteristics affected more and others less (e.g., positive emotional characteristics more, negative less)? In particular, what are the effects of reverberation time and amount? What are the effects of hall size and listener position? Which instruments sound emotionally stronger to listeners in the front or back of small and large halls? Are dry sounds without reverberation emotionally dry as well, or, do they have distinctive emotional characteristics?

2. BACKGROUND

2.1 Music Emotion and Timbre

Researchers have considered music emotion and timbre together in a number of studies, which are well-summarized in [5].

2.2 Reverberation

2.2.1 Artificial Reverberation Models

Various models have been suggested for reverberation using different methods to simulate the build-up and decay of reflections in a hall such as simple reverberation algorithms using several feedback delays [6], simulating the time and frequency response of a hall [7, 8, 9, 10], and convolving the impulse response of the space with the audio signal to be reverberated [11, 12]. They can be characterize by Reverberation time (RT_{60}) which measures the time reverberation takes to decay by 60dB SPL from an initial impulse [13].

2.2.2 Reverberation and Music Emotion

Västfjäll et al. [14] found that long reverberation times were perceived as most unpleasant. Tajadura-Jiménez et al. [15] suggested that smaller rooms were considered more pleasant, calmer, and safer than big rooms, although these differences seemed to disappear for threatening sound sources. However, it is still largely unknown to what extent reverberation influences the emotional characteristics of musical instrument sounds.

3. METHODOLOGY

3.1 Overview

To address the questions raised in Section 1, we conducted a listening test to investigate the effect of reverberation on the emotional characteristics of individual instrument sounds. We tested eight sustained musical instruments including bassoon (bs), clarinet (cl), flute (fl), horn (hn), oboe (ob), saxophone (sx), trumpet (tp), and violin (vn). The original anechoic sounds were obtained from the *University of Iowa Musical Instrument Samples* [16]. They had fundamental frequencies close to Eb4 (311.1 Hz), and were analyzed using a phase-vocoder algorithm [17]. We resynthesized the sounds by additive sinewave synthesis at exactly 311.1 Hz, and equalized the total duration to 1.0s. Loudness of the sounds were also equalized by manual adjustment.

We compared the anechoic sounds with reverberation lengths of 1s and 2s. The reverberation generator provided by *Cool Edit* [18] was used in our study. Its "Concert Hall Light" preset is a reasonably natural sounding reverberation. This preset uses 80% for the amount of reverberation corresponding to the back of the hall, and we approximated the front of the hall with 20%. Thus, in addition to the anechoic sounds, there were four reverberated sounds for each instrument.

34 subjects without hearing problems were hired to take the listening test. All subjects were fluent in English. They compared the stimuli in paired comparisons for eight emotional categories: Happy, Sad, Heroic, Scary, Comic, Shy, Romantic, and Mysterious. Some choices of emotional characteristics are fairly universal and occur in many previous studies roughly corresponding to the four quadrants of the Valence-Arousal plane [19]. In the listening test, every subject heard paired comparisons of all five types of reverberation for each instrument and emotional category. During each trial, subjects heard a pair of sounds from the same instrument with different types of reverberation and were prompted to choose which more strongly aroused a given emotional category. Each permutation of two different reverberation types were presented, and the listening test totaled $P_2^5 \times 8 \times 8 = 800$ trials. For each instrument, the overall trial presentation order was randomized (i.e., all the bassoon comparisons were first in a random order, then all the clarinet comparisons second, etc.). The listening test took about 2 hours, with breaks every 30 minutes.

4. LISTENING TEST RESULTS

We ranked the tones by the number of positive votes they received for each instrument and emotional category, and derived scale values using the Bradley-Terry-Luce (BTL) statistical model [21, 22]. The BTL value is the probability that listeners will choose that reverberation type when considering a certain instrument and emotional category. For each graph, the BTL scale values for the five tones sum up to 1. Therefore, if all five reverb types were judged equally happy, the BTL scale values would be 1/5 = 0.2.

Figures 1 to 5 show BTL scale values and the corresponding 95% confidence intervals for each emotional category and instrument. Based on Figures 1 - 5, Table 1 shows the number of times each reverb type was significantly greater than the other four reverb types (i.e., where the bottom of its 95% confidence interval was greater than the top of their 95% confidence interval) over the eight instruments. Table 1 shows the maximum value for each emotional category in bold in a shaded box (except for Shy since all its values are zero or near-zero).

Table 1 shows that for the emotional category Happy, Small Hall had most of the significant rankings. This result agrees with that found by Tajadura-Jiménez [15], who found that smaller rooms were most pleasant. The result also agrees with Västfjäll [14], who found that larger reverberation times were more unpleasant than shorter ones. However, for Heroic, our finding was in contrast to that found by Västfjäll and Tajadura-Jiménez. As Heroic is also high-Valence, they would have predicted that Heroic would have had a similar result as Happy. Though Large Hall Back was ranked significantly greater more often than all the other options combined.

Table 1 also shows that Anechoic was the most Comic, while Large Hall Back was the least Comic. This basically agrees with Västfjäll [14] and Tajadura-Jiménez [15].

Large Hall Back was the most Sad in Table 1 (though Small Hall Back and Large Hall Front were not far behind). Large Hall Back was more decisively on top for Scary. Since Sad and Scary are both low-Valence, these results agree with Västfjäll [14] and Tajadura-Jiménez [15] who found that larger reverberation times and larger rooms were more unpleasant. Reverb had very little effect on Shy in Table 1.

The Romantic rankings in Figure 5 were more widely spaced than the other categories, and Table 1 indicates that Large Hall Back was significantly more Romantic than most other reverb types. Like Heroic, this result is in contrast to the results of Västfjäll [14] and Tajadura-Jiménez [15] since Romantic is high-Valence. The bassoon for Romantic was the most strongly affected among all instruments and emotional categories. Similar to Romantic, the Mysterious rankings were also widely spaced.

In summary, our results show distinctive differences between the high-Valence emotional categories Happy, Heroic, Comic, and Romantic. In this respect, our results contrast with the results of Västfjäll [14] and Tajadura-Jiménez [15].

5. DISCUSSION

Based on Table 1, our main findings are the following:

1. Reverberation had a strongly significant effect on Mysterious and Romantic for Large Hall Back.
2. Reverberation had a medium effect on Sad, Scary, and Heroic for Large Hall Back.
3. Reverberation had a mild effect on Happy for Small Hall Front.
4. Reverberation had relatively little effect on Shy.
5. Reverberation had an opposite effect on Comic, with listeners judging anechoic sounds most Comic.

Table 1 shows very different results for the high-Valence emotional categories Happy, Heroic, Comic, and Romantic. The results of Västfjäll [14] and Tajadura-Jiménez [15] suggested that all these emotional characteristics would be stronger in smaller rooms. Only Happy and Comic were stronger for Small Hall or Anechoic, while Heroic and Romantic were stronger for Large Hall. The above results give audio engineers and musicians an interesting perspective on simple parametric artificial reverberation.

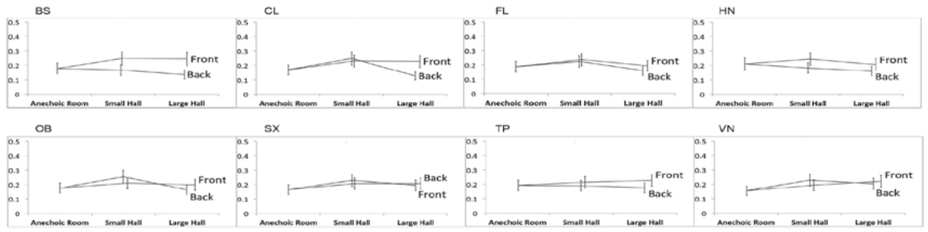


Figure 1. BTL scale values and the corresponding 95% confidence intervals for the emotional category Happy.

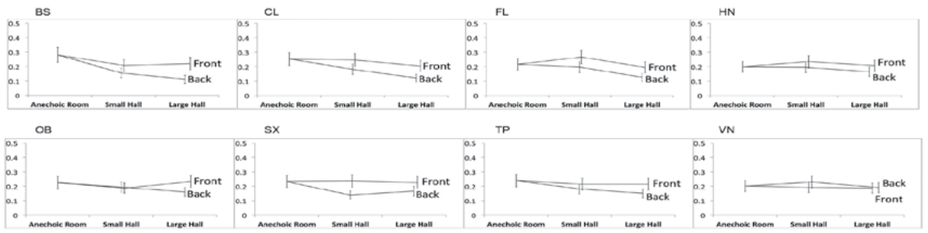


Figure 2. BTL scale values and the corresponding 95% confidence intervals for Comic.

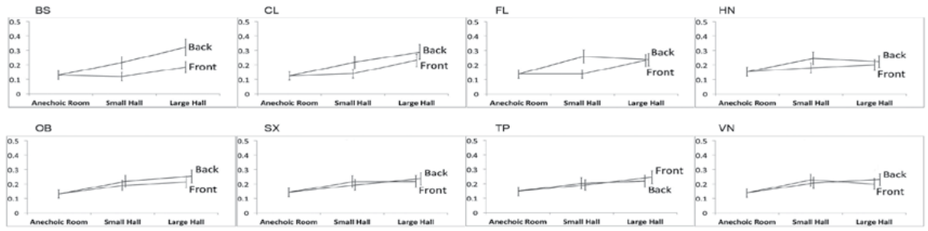


Figure 3. BTL scale values and the corresponding 95% confidence intervals for Sad.

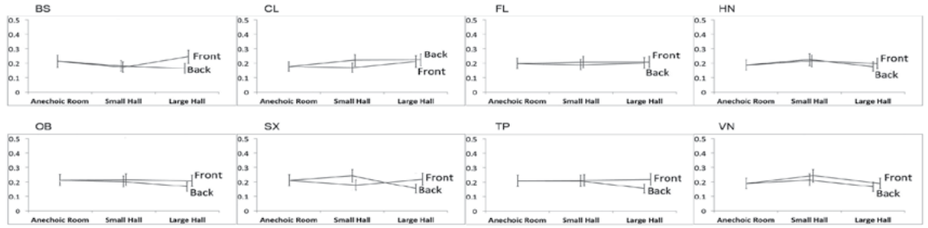


Figure 4. BTL scale values and the corresponding 95% confidence intervals for Shy.

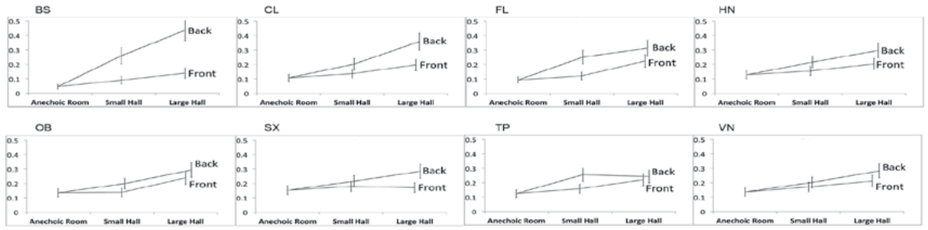


Figure 5. BTL scale values and the corresponding 95% confidence intervals for Romantic.

Emotion Category \ Reverb Type	Anechoic	Small Hall Front	Small Hall Back	Large Hall Front	Large Hall Back
Happy	0	**4**	**3**	**2**	0
Heroic	0	0	**3**	**2**	**7**
Comic	**6**	4	1	4	0
Sad	0	2	9	7	**11**
Scary	0	1	5	4	**11**
Shy	0	0	1	0	0
Romantic	0	1	9	9	**23**
Mysterious	0	1	12	7	**29**

Table 1. How often each reverb type was statistically significantly greater than the others over the eight instruments.

6. REFERENCES

[1] T. Eerola, R. Ferrer, and V. Alluri, "Timbre and Affect Dimensions: Evidence from Affect and Similarity Ratings and Acoustic Correlates of Isolated Instrument Sounds," *Music Perception: An Interdisciplinary Journal*, vol. 30, no. 1, pp. 49–70, 2012.

[2] B. Wu, A. Horner, and C. Lee, "Musical Timbre and Emotion: The Identification of Salient Timbral Features in Sustained Musical Instrument Tones Equalized in Attack Time and Spectral Centroid," in *International Computer Music Conference (ICMC), Athens, Greece*, 14-20 Sept 2014, pp. 928–934.

[3] C.-j. Chau, B. Wu, and A. Horner, "Timbre Features and Music Emotion in Plucked String, Mallet Percussion, and Keyboard Tones," in *International Computer Music Conference (ICMC), Athens, Greece*, 14-20 Sept 2014, pp. 982–989.

[4] B. Wu, C. Lee, and A. Horner, "The Correspondence of Music Emotion and Timbre in Sustained Musical Instrument Tones," *Journal of the Audio Engineering Society*, vol. 62, no. 10, pp. 663–675, 2014.

[5] C.-j. Chau, B. Wu, and A. Horner, "The Emotional Characteristics and Timbre of Nonsustaining Instrument Sounds," *Journal of the Audio Engineering Society*, vol. 63, no. 4, pp. 228–244, 2015.

[6] M. R. Schroeder, "Natural sounding artificial reverberation," *Journal of the Audio Engineering Society*, vol. 10, no. 3, pp. 219–223, 1962.

[7] ——, "Digital simulation of sound transmission in reverberant spaces," *The Journal of the Acoustical Society of America*, vol. 47, no. 2A, pp. 424–431, 1970.

[8] J. A. Moorer, "About this reverberation business," *Computer Music Journal*, pp. 13–28, 1979.

[9] J.-M. Jot and A. Chaigne, "Digital delay networks for designing artificial reverberators," in *90th Audio Engineering Society Convention*. Audio Engineering Society, 1991.

[10] W. G. Gardner, "A realtime multichannel room simulator," *J. Acoust. Soc. Am*, vol. 92, no. 4, p. 2395, 1992.

[11] A. Reilly and D. McGrath, "Convolution processing for realistic reverberation," in *98th Audio Engineering Society Convention*. Audio Engineering Society, 1995.

[12] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *108th Audio Engineering Society Convention*. Audio Engineering Society, 2000.

[13] W. C. Sabine and M. D. Egan, "Collected papers on acoustics," *The Journal of the Acoustical Society of America*, vol. 95, no. 6, pp. 3679–3680, 1994.

[14] D. Västfjäll, P. Larsson, and M. Kleiner, "Emotion and auditory virtual environments: affect-based judgments of music reproduced with virtual reverberation times," *CyberPsychology & Behavior*, vol. 5, no. 1, pp. 19–32, 2002.

[15] A. Tajadura-Jiménez, P. Larsson, A. Väljamäe, D. Västfjäll, and M. Kleiner, "When room size matters: acoustic influences on emotional responses to sounds," *Emotion*, vol. 10, no. 3, pp. 416–422, 2010.

[16] "University of Iowa Musical Instrument Samples," *University of Iowa*, 2004, http://theremin.music.uiowa.edu/MIS.html.

[17] J. W. Beauchamp, "Analysis and synthesis of musical instrument sounds," in *Analysis, Synthesis, and Perception of musical sounds*. Springer, 2007, pp. 1–89.

[18] "Cool Edit," *Adobe Systems*, 2000, https://creative.adobe.com/products/audition.

[19] P. N. Juslin and J. Sloboda, *Handbook of music and emotion: Theory, research, applications*. Oxford University Press, 1993.

[20] R. A. Bradley, "14 Paired comparisons: Some basic procedures and examples," *Nonparametric Methods*, vol. 4, pp. 299–326, 1984.

[21] F. Wickelmaier and C. Schmid, "A Matlab Function to Estimate Choice Model Parameters from Paired-comparison Data," *Behavior Research Methods, Instruments, and Computers*, vol. 36, no. 1, pp. 29–40, 2004.

General Characteristics Analysis of Chinese Folk Songs based on Layered Stabilities Detection(LSD) Audio Segmentation Algorithm

Juan Li
lijuan@mail.xjtu.edu.cn

Yinrui Wang
Xi'an Jiaotong University
yr.wang@stu.xjtu.edu.cn

Xinyu Yang
yxyphd@mail.xjtu.edu.cn

ABSTRACT

This paper discusses a method to automatically analyze the general characteristics of Chinese folk songs. This not only makes it possible to find general characteristics from a large number of folk songs, but it also provides people with a more profound understanding of the creation of them. We use the styles of folk songs' music structure types we proposed in each region to study the general characteristics. The process consists of three steps: first, segment each folk song into clips based on LSD audio segmentation algorithm we first proposed. Then, music structure annotation to these clips. Finally, make statistics on the styles of each folk song's music structure types and analyze their general characteristics. Experiments show that it is feasible to automatically analyze the general characteristics of folk songs based on the styles of music structure types we proposed. The general characteristics of the folk songs in three regions are based on the reality that all music structure types and styles have similar ratios, the Coordinate Structure has the most, and the Cyclotron Structure has the least.

1. INTRODUCTION

In recent years, with the expansion of media, especially the rapid development of the Internet, Chinese folk songs have begun to be concerned, liked and studied by more and more people. At the same time, inspired by biological gene, more and more scholars have begun to pay close attention to the nature and the common musical attributes of them. However, due to their huge number and diversity, it is very difficult to get the general characteristics of them directly through the artificial statistics. So, it becomes very meaningful and important to find a method to automatically analyze the general characteristics of folk song.

Chinese folk song is an important genre of Chinese national folk music that was produced and developed through extensive oral singing. Folk songs in different regions have clear differences caused by lifestyle and living environment, but as a result of the combination of the characteristics of different regions' folk songs through the process of communication, they also have very strong similarities. Based on the "similarities", music information retrieval systems and the Chinese folk music field have studied the general characteristics of folk songs in last couple years [1–4]. In a study of the general characteristics analysis of Chinese folk songs, [1] pointed out that Chinese folk songs mostly belong to the Chinese music system, the pentatonic and heptatonic scales are the most common. [2] argued that the folk songs in the Northeast Plain of China have a genetic relationship that shows consistency in the aspects of genre and melody style. [3] suggested that "opening, developing, changing and concluding" structure is very common in Chinese folk songs, and the most frequently used is the form structure of the upper and lower sentence. [4] showed that the majority of the texture form of Chinese folk songs is monophonic, and the music mode is mainly major.

However, existing research methods on the general characteristics analysis generally rely on artificial statistical music attributes of different songs. This will not only introduce errors, but the statistics will become very difficult when the data set is particularly large. In addition, these general characteristics studies cannot provide a concrete way to obtain the general characteristics. Although people who have a general understanding of music can sense the general characteristics of folk songs through hearing, they generally have no more than a vague feeling and lack a method to distinguish and identify them correctly.

In view of the problems in the existing research general characteristics analysis of folk songs, this paper tried to discover a method to automatically analyze the general characteristics of Chinese folk songs. We use the styles of folk songs' music structure types we proposed in each region to study the general characteristics. The process consists of three steps: first, segment each folk song into clips based on LSD audio segmentation algorithm we proposed. Then, music structure annotation to these clips. Finally, make statistics on the styles of each folk song' music structure types and analyze their general characteristics. The experimental results show that it is feasible to study the general characteristics of folk songs based on the music structure. The general characteristics of the folk songs in three region is that all music structure types and styles have similar ratios, the Coordinate Structure has the most, and the Cyclotron Structure has the least.

The article is organized as follows: In Section 2, we propose the LSD audio segmentation algorithm. In Section 3, we use the music structure to analyze the general characteristics of Chinese folk songs. Experiments and results are described in 4, which is structured as follows: The data set and experimental setup are introduced in Section 4.1. In Section 4.2, we demonstrate the effectiveness of the LSD audio segmentation algorithm. The feasibility of the general characteristics analysis is introduced in Section 4.3. Finally, conclusions are drawn in Section 5.

2. LSD AUDIO SEGMENTATION ALGORITHM

To obtain the styles of each folk song's music structure types, first of all, we should get the music structure of each folk song. This consists of two steps, segment each folk song into clips according to music similarity and music annotation to the clips of each folk song. This section, we proposed a new audio segmentation algorithm, the LSD audio segmentation algorithm, to segment each folk song. In subsection 2.1, the principle of detecting the acoustic change points according to the changing trend of stabilities is described in detail. Subsection 2.2 gives an account of the detailed process of the LSD algorithm.

2.1 Principle of acoustic change points detection based on the stabilities changing trend

We detect the acoustic change points based on the changing trend of stabilities. The principle is as follows.

The features,MFCC, LPCC [5], LSP [6], Tempo, FP [7], Chroma [8], SFM, SCF [9],are extracted by frame for each song. These features are recorded as $X_1 = \{x_1, x_2, ..., x_N\}$, where N is the number of audio frames and the dimension of each frame features is P. Suppose $x_k \in X$ is an acoustic change points, then the audio features X are divided into two parts,$X_1 = \{x_1, x_2, ..., x_k\}$ and $X_2 = \{x_{k+1}, x_{k+2}, ..., x_N\}$, by x_k, assuming that the two parts respectively obey the $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$ distributions, then we give the follow definition.

Definition 1 *The stability $ST(x_k)$ is the log-likelihood sum of two part signals,on the left and right sides of x_k, over their respective distributions, that is:*

$$\begin{aligned} ST(x_k) &= L(X_1|N(\mu_1,\Sigma_1)) + L(X_2|N(\mu_2,\Sigma_2)) \\ &= \sum_{j=1}^{k} \lg P[x_j|N(\mu_1,c)] + \sum_{j=k+1}^{N} \lg P[x_j|N(\mu_2,\Sigma_2)] \\ &= \frac{\mathrm{PN}}{2}\lg 2\pi - \frac{\mathrm{k}}{2}\lg|\Sigma_1| - \frac{\mathrm{N-k}}{2}\lg|\Sigma_2| - \\ &\quad \frac{1}{2}\sum_{j=1}^{k} A^T\Sigma_1^{-1}A - \frac{1}{2}\sum_{j=k+1}^{N} B^T\Sigma_2^{-1}B \end{aligned} \tag{1}$$

where, $A = x_j - \mu_1$,$B = x_j - \mu_2$. Therefore, theorem 1 will hold when calculating the stability of each frame.

Theorem 1 *On the left side of the acoustic change point x_k, the stabilities shows an upward tendency as the audio frame approaches x_k; On the right side of the acoustic change point x_k, the stabilities shows a decreasing tendency as the audio frame moves away from x_k. The stability will achieve its maximum value at the acoustic change point x_k.*

Proof 1 *Assuming that x_m and x_{m+1} are two adjacent points on the left side of the acoustic change point x_k, then there will be $\lg P[x_{m+1}|N(\mu_1,\Sigma_1)] > \lg P[x_{m+1}|N(\mu_2,\Sigma_2)]$. Hence, we can obtain:*

$$\begin{aligned} &ST(x_{m+1}) - ST(x_m) = \\ &\sum_{j=1}^{m+1} \lg P[x_j|N(\mu_1,\Sigma_1)] + \sum_{j=m+2}^{N} \lg P[x_j|N(\mu_2,\Sigma_2)] \\ &- (\sum_{j=1}^{m} \lg P[x_j|N(\mu_1,\Sigma_1)] + \sum_{j=m+1}^{N} \lg P[x_j|N(\mu_2,\Sigma_2)]) \\ &= \lg P[x_{m+1}|N(\mu_1,\Sigma_1)] - \lg P[x_{m+1}|N(\mu_2,\Sigma_2)] > 0 \end{aligned} \tag{2}$$

That is, on the left side of the acoustic change point x_k, the stabilities show an upward tendency. In the same way, we can prove that the stabilities show a decreasing tendency on the right side of the acoustic change point x_k.

From the theorem 1, we can obtain that if we want to judge whether there is a acoustic change point in the audio fragment or a frame is the actual acoustic change point, we should get the distributions on both sides of the actual acoustic change point. But we don't even know where the acoustic change point is, how to know the distribution of its two sides. obviously, we can use the following method to approximate computing the distributions on both sides of the acoustic change point: first cluster the frames of the audio fragment into two classes using the K-means algorithm [10], and then use the mean and variance of the two classes as the parameters of Equation (1). Therefore we can detect the acoustic change points based on the changing trend of stabilities.

2.2 LSD audio segmentation algorithm

According to Theorem 1, we can obtain that some acoustic change points will be undetected when the sliding window is too large. To avoid this, we adopt the method of top-down layered detection. The LSD audio segmentation algorithm consists of two processes: 1) Layered detection of acoustic change points according to Theorem 1 in a sliding window. 2) Acoustic change points detected of the whole audio.

2.2.1 Layered detection of acoustic change points in a sliding window

Fig. 1 shows the process of the layered detection of acoustic change points, according to Theorem 1, in a sliding window W_0.

Because of using the method of top-down layered detection, the length $N_{\min}$ of the minimum analysis window W_m should be first determined before detecting the acoustic change points. we assume that there is only one acoustic change point in an analysis window every time. The specific process is as follows:

(i) Extract audio features by frame and determine the length $N_{\min}$ of the minimum analysis window W_m.

(ii) Calculate the stability of each frame using Equation (1), and then select the frame *k* with a maximum stability as the pre-selection acoustic change point. Then, determine whether the pre-selection acoustic change point is true or not according to Theorem 1. To ensure that there is enough data to make the stability calculation more reliable, the stability of $N_M (0 < 2N_M < N_{\min})$ frames at the beginning and end of the window can be not calculated. Equation (3) can be used to judge the stabilities obey Theorem 1.

$$\begin{aligned} &IncNum_L(k) > \alpha \cdot Num_L \\ &SumIncST_L(k) > SumDecST_L(k) \\ &DecNum_R(k) > \alpha \cdot Num_R \\ &SumDecST_R(k) > SumIncRT_R(k) \end{aligned} \tag{3}$$

$IncNum_L(k)$ is the total increasing times of stabilities, Num_L is the number of frames, $SumIncST_L(k)$ is the total increasing amount of the stabilities, and $SumDecST_L(k)$ is the total decreasing amount of the stabilities on the left side of frame *k*. $DecNum_R(k)$ is the total decreasing times of the stabilities, Num_R is the number of

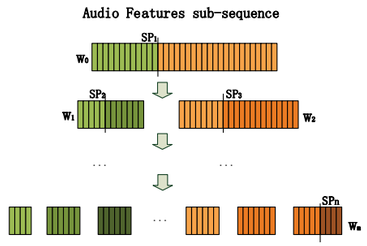


Figure 1. Process of layered detection of acoustic change points in a sliding window W_0

frames, $SumIncST_R(k)$ is the total increasing amount of the stabilities, and $SumDecST_R(k)$ is the total decreasing amount of the stabilities on the right side of frame k.α is the percent of the audio frames. The reason for choosing the stabilities increasing and decreasing times is to eliminate the influence of instantaneous drastic changes in the stabilities. The choice of the total increasing amount and total decreasing amount of the stabilities is to solve the problem that the increasing and decreasing times are inconsistent with the total increasing and decreasing amount of the stabilities.

(iii) If Equation (3) is not set up, according to Theorem 1, we can conclude that the analysis window does not contain an acoustic change point.

(iv) If Equation (3) is set up, the pre-selection acoustic change point, frame k, is true acoustic change point, and it is placed into the acoustic change point set CP. Then, the window is divided into two sub windows using the acoustic change point as the boundary, and it is determined whether the length of the sub window is less than the minimum window $N_{\min}$. It will not be dealt with if it is less than the length of the minimum window $N_{\min}$; on the contrary, sub windows continue to execute (ii) step.

2.2.2 Acoustic change points detection of a whole audio

It is easy to obtain all acoustic change points of a whole song based on the process 1). First determine the length $N_{\max}$ of the sliding window W_0 and then place the sliding analysis window W_0 at the starting position of the feature sequence of folk songs. If no acoustic change point is detected, move the window backwardΔl $(0 < \Delta l < N_{\min})$ distance. If it is detected, a sequence of acoustic change points with a significant degree of sorting is obtained in a sliding window Move the window to the acoustic change point $SP_l^{tl_{\max}}$ with a maximum time label, and then continue to detect the next sliding window. After the sliding window traverses the entire audio stream of folk songs, the set CP of all of the acoustic change points will be obtained. Then, sort set CP according to the time tag. Finally, we can segment the audio based on set CP.

After each folk song is segmented,the features of MFCC, LPCC, LSP, Tempo, FP, Chroma, SFM, and SCF are re-extracted, using each clip of each folk song as a whole. Then, we use these re-extracted features of each folk song to annotate the music structure of them.

3. GENERAL CHARACTERISTICS ANALYSIS OF FOLK SONGS

Each song is segmented into clips based on LSD audio segmentation algorithm in section 2.2, so the music structure of each folk song can be obtained by music structure annotation. The process of generating the music structure of a song is shown in Fig. 2.

In Fig. 2, first of all, cluster the clips and annotate different classes with different labels. Then, the audio clips are corresponding to respective class labels according to the time sequence, and each clip is recorded as a triad $(Tag, T_{start}, T_{end})$ when implemented, where Tag is a class label, T_{start} is the start time of the clip and T_{end} is the end time of the clip. The music structure of the song in Fig. 2 is **ABABABCABC**. In the process, we use the agglomerate hierarchical clustering algorithm [11] to cluster the re-extracted features for each folk song, and the silhouette coefficient [12] is chosen as a measurement standard to determine the cluster number. We select the number with the minimum silhouette coefficient.

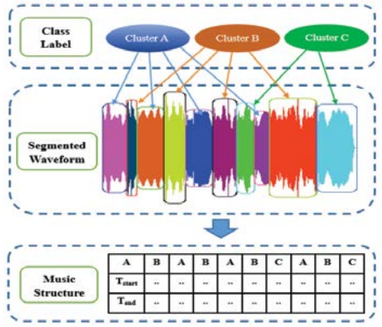


Figure 2. The annotation of music structure

In this paper, the music structure is classified into four types, named coordinate structure, reproducing structure, circular structure, and cyclotron structure, according to the order of different labels in the music structure of a song. The concrete forms of the four music structure types are as follows:

Coordinate Structure: a music structure type shaped like the style "A", "A+B", "A+B+C",or "A+B+C+D" in which the labels of all clips are different.

Reproducing Structure: a music structure type shaped like the style "A+B+A", where two identical labels are inserted around a different label.

Circular Structure: a music structure type shaped like the style "A+B+A+B" or "A+B+C+A+B+C", where a group of labels recurs.

Cyclotron Structure: a music structure type shaped like the style "A+B+A+C+A" or "A+B+A+C+A+D", where a label appears at least two times and the labels in front of it label and behind it are different.

Musical melody has the principles of change, contrast and repetition. Reflected in the music structure, it means that the music structures of folk songs have a coordinate structure, reproducing structure, circular structure, cyclotron structure or other music structure types. The variety of the styles of all music structure types can be more easily implemented in general characteristics analysis.

The general characteristics analysis of folk songs prepares statistics on the styles of all music structure types and then analyze their similarities. The statistical styles of music structure types should follow the priority of the styles of cyclotron structure, the circular structure, the reproducing structure and the coordinate structure. For example, if the music structure of a song is **ABCBABABA**, we can first identify the second label to the sixth label as **BCBAB**, the style of the cyclotron structure, then obtain the style of the reproducing structure from the seventh label to the ninth label **ABA**, and finally, the style of the coordinate structure, the first label, is obtained.

4. EXPERIMENT ON THE GENERAL CHARACTERISTICS ANALYSIS OF FOLK SONGS

The previous sections of this paper theoretically analyze the significance of the general characteristics analysis of Chinese folk songs, and to obtain the music structure, the LSD audio segmentation algorithm and the process of music structure annotation are introduced. It then puts forward the automatic analysis of the folk songs' general characteristics based on the styles of their music structure types. In this section, we will prove the effectiveness of the LSD audio segmentation algorithm and the feasibility of the general characteristics analysis.

4.1 Dataset and Experimental Setup

We select representative Chinese folk songs, XinTianYou-Shaanxi, XiaoDiao-Jiangnan, and Haozi-Hunan, as our data set in the experiment. They also represent different styles (mountain songs, minor and work songs) of Chinese folk songs that can make the general characteristics discovered more convincing. The datasets are derived from "Integration of Chinese folk songs" [13], the numbers of the folk songs in the three regions are 109,101, and 134, respectively.

In sub experiment to prove the effectiveness of the LSD audio segmentation algorithm, we randomly select 100 folk songs that have known the acoustic change points from the three regions. The frame length is 20 ms, the frame shift is 10 ms, the size of sliding window is 1600 frames, the size of the minimum window is 200 frames, the sliding window moving distance is 50 frames, and is 75%.

4.2 Effectiveness of the LSD audio segmentation algorithm

To measure the performance of the LSD audio segmentation algorithm, we use false alarm (FA), recall (RCL), precision (PRC), and F-measure, respectively. First, we define some variables:

NW: the number of wrongly detected acoustic change points. *NC*: the number of correctly detected acoustic change points. *NT*: the total number of true acoustic change points. *ND*: the total number of detected acoustic change points, where *ND=NW+NC*.

FA, RCL, PRC and F-measure(F) are defined as:

$$\begin{aligned} FA &= \frac{NW}{NT+NW} \times 100\% \\ RCL &= \frac{NC}{NT} \times 100\% \\ PRC &= \frac{NC}{ND} \times 100\% \\ F &= 2 \times \frac{PRC \times RCL}{PRC + RCL} \times 100\% \end{aligned} \tag{4}$$

The averaged detection results for the LSD audio segmentation algorithms are shown in Fig. 3.

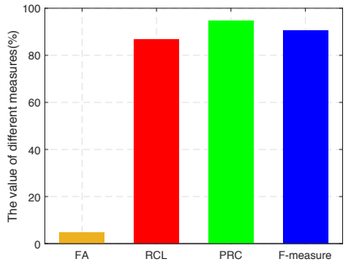


Figure 3. The averaged detection results of the LSD algorithms

Fig. 3 shows that, the RCL(86.64%), PRC(94.53%) and F-measure(90.39%) are large. This is consistent with the analysis in Section 2, the LSD algorithm detects the acoustic change points based on the changing trend of stabilities, which is well consistent with the variation of acoustic features around the acoustic change point, so the RCL, PRC, and F-measure is large. In addition, due to the noise and homophonic in folk songs, some frames are mistaken for acoustic change point and then are wrongly detected, but from Fig. 3, we can obtain that the FA(4.78%) of the LSD audio segmentation algorithm is small, it's almost has no effect on audio segmentation. Therefore, we can get the conclusion that the LSD audio segmentation algorithm we proposed is effective.

4.3 Feasibility of the general characteristics analysis of Chinese folk songs

In according with the music structure annotation method introduced in Section 3, we can obtain the music structure of every folk song and then can produce statistics on the number and the proportion of all styles of music structure types. The statistical results of the three regions are respectively shown in Table 1, Table 2 and Table 3.

Music Structure Type	Music Structure Style	Proportion (%)	Total (%)
Coordinate Structure	A	17.28	79.58
	A+B	24.47	
	A+B+C	3.66	
	A+B+C+D+...	37.17	
Reproducing Structure	A+B+A	12.04	12.04
Cyclotron Structure	A+B+A+C+A	0.52	0.52
	A+B+A+C+A+...	0	
Circular Structure	A+B+A+B	2.62	7.85
	A+B+A+B+A+B	2.62	
	A+B+A+B+A+B+...	2.62	

Table 1. Statistical results of XinTianYou-Shaanxi music structure styles

Music Structure Type	Music Style	Proportion (%)	Total (%)
Coordinate Structure	A	20.88	79.67
	A+B	19.78	
	A+B+C	3.30	
	A+B+C+D+...	35.71	
Reproducing Structure	A+B+A	10.99	10.99
Cyclotron Structure	A+B+A+C+A	0.55	0.55
	A+B+A+C+A+...	0	
Circular Structure	A+B+A+B	3.85	8.79
	A+B+A+B+A+B	1.65	
	A+B+A+B+A+B+...	3.30	

Table 2. Statistical results of XiDiao-Jiangnan music structure styles

Music Structure Type	Music Style	Proportion (%)	Total (%)
Coordinate Structure	A A+B A+B+C A+B+C+D+...	17.50 20.42 3.33 34.17	75.42
Reproducing Structure	A+B+A	10.00	10.00
Cyclotron Structure	A+B+A+C+A A+B+A+C+A+...	1.25 0	1.25
Circular Structure	A+B+A+B A+B+A+B+A+B A+B+A+B+A+B+...	5.00 1.67 6.25	12.92

Table 3. Statistical results of HaoZi-Hunan music structure styles

The last columns of Table 1, Table 2 and Table 3 all show that the folk songs in the three regions have general characteristics: the coordinate structure occupies the largest proportion, and the cyclotron structure the least. The reason why the coordinate structure is the most common is that it is the simplest combination of the music structure and is the foundation of all music structure types. On the other hand, the strict requirements for the formation of the cyclotron structure make it the least common. It needs to have two inconsistent clips in three adjacent clips with the same label, which leads to its not being stable and easily transitioning to the reproducing structure and circular structure.

We also compare the proportions of each music structure styles in the three region' folk songs from Table 1, Table 2, and Table 3, We can see another indication of the general characteristics of the three region folk songs, as all the music structure styles have similar ratios.

In conclusion, we can identify the general characteristics of the three regions' folk songs are that, they have strong similarities in the music structure types and styles, having similar ratios,with the coordinate Structure the most and the cyclotron structure the least.

5. CONCLUSIONS

This paper studies the general characteristics of Chinese folk songs using the styles of folk songs' music structure types. The process consists of three steps: first, segment each folk song into clips based on LSD audio segmentation algorithm we proposed. Then, music structure annotation to these clips. Finally, make statistics on the styles of each folk song' music structure types and analyze their general characteristics.

The experiments show that the LSD audio algorithm we proposed is effective for audio segmentation according to music similarity. The F-measure can reach 90.39%. It is feasible to automatically analyze the general characteristics of folk songs based on the music structure types we proposed, and the general characteristics of the three regions' folk songs is that all the music structure types and styles have similar ratios, with the coordinate structure being the most and the cyclotron structure the least.

6. ACKNOWLEDGEMENT

The work is supported in part by the fundamental research funds for the central universities: sk2016017. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

7. REFERENCES

[1] L. K, "Theme Motif Analysis as Applied in Chinese Folk Songs," *Art of Music-Journal of the Shanghai Conservatory of Music*, 2005.

[2] Z.-Y. P, "The Common Characteristics of Folk Songs in Plains of Northeast China," *Journal of Jilin College of the Arts*, 2010.

[3] Y. Ruiqing, "Chinese folk melody form (22)," *Journal of music education and creation*, vol. 5, pp. 18–20, 2015.

[4] Z. Shenghao, "The interpretation of Chinese folk songs and music ontology elements," *Ge Hai*, vol. 4, pp. 48–50, 2013.

[5] G. Mantena, S. Achanta, and K. Prahallad, "Query-by-example spoken term detection using frequency domain linear prediction and non-segmental dynamic time warping," *IEEE/ACM Trans. Audio, Speech, Lang.Process*, vol. 22, no. 5, pp. 946–955, 2014.

[6] G. Min, X. Zhang, J. Yang, and Y. Chen, "Sparse representation and performance analysis for LSP parameters via dictionary learning," *Journal of Pla University of Science & Technology*, 2014.

[7] D. Bogdanov, J. Serrà, N. Wack, P. Herrera, and X. Serra, "Unifying low-level and high-level music similarity measures," *IEEE Trans.Multimedia*, vol. 13, no. 4, pp. 687–701, 2011.

[8] M. Mller and S. Ewert, "Chroma Toolbox: Matlab Implementations for Extracting Variants of Chroma-Based Audio Features." in *in Proc. of ISMIR International Society for Music Information Retrieval Conference*, 2011, pp. 215–220.

[9] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Trans.Multimedia*, vol. 13, no. 2, pp. 303–319, 2011.

[10] B. K. Mishra, A. Rath, N. R. Nayak, and S. Swain, "Far efficient K-means clustering algorithm," in *Proc. of ACM International Conference on Advances in Computing, Communications and Informatics*, 2012, pp. 106–110.

[11] Y. Tamura and S. Miyamoto, "A method of two stage clustering using agglomerative hierarchical algorithms with one-pass k-means++ or k-median++," in *Proc. of IEEE Granular Computing*, 2014, pp. 281–285.

[12] R. Etemadpour, R. Motta, J. G. de Souza Paiva, R. Minghim, F. de Oliveira, M. Cristina, and L. Linsen, "Perception-based evaluation of projection methods for multidimensional data visualization," *IEEE Trans.Vis Comput Grap.*, vol. 21, no. 1, pp. 81–94, 2015.

[13] T. E. C. of "Integration of Chinese folk songs", *Integration of Chinese folk songs*. Chinese ISBN center, 1994.

Additive Synthesis with Band-Limited Oscillator Sections

Peter Pabon
Institute of Sonology,
Royal Conservatoire, Juliana van Stolberglaan 1,
2595 CA Den Haag, The Netherlands
pabon@koncon.nl

So Oishi
Institute of Sonology,
Royal Conservatoire, Juliana van Stolberglaan 1,
2595 CA Den Haag, The Netherlands
oishiso@gmail.com

ABSTRACT

The band-limited oscillator (BLOsc) is atypical as it produces signal spectra with distinctive edgings instead of distinct peaks. An edging at low frequency can have a comparable perceptual effect as a spectral peak. When modulated, the BLOsc has the advantage that it preserves spectral textures and contrasts that tend to blur with a resonance-based (subtractive) synthesis approach. First, the simple math behind the BLOsc is described. Staying close to this formulation helps to keep the model malleable and to maintain the dynamic consistencies within the model. Next, an extended processing scheme is presented that essentially involves a sectioned evaluation of the frequency range. The modulation and the application of convolution-, and chance-mechanisms are examined. Stochastic control, MFCC based control and the options of formant modeling are shortly discussed. Implementations in MAX/MSP and Super Collider are used to demonstrate the different options.

1. INTRODUCTION

Before digital became the leading approach in electronic sound synthesis, Moorer [1] introduced the band-limited oscillator (BLOsc) principle as a means to synthesize complex audio spectra with only a limited set of frequency-coupled oscillators. At the time in 1976 the technique was still called "discrete summation". The BLOsc uses an efficient calculation scheme to synthesize signals that, over a hard-limited harmonic range, show a constant exponentially varying spectrum envelope, one that is smooth when measured on a scale in dB/harmonic.

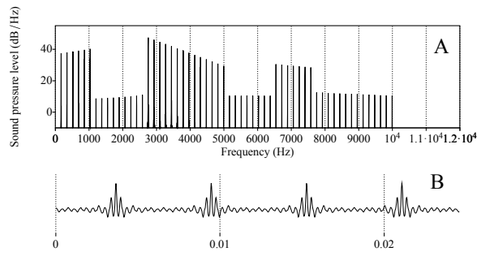


Figure 1. Sectioned BLOsc, (A) spectrum & (B) signal.

Expanding on this principle, the harmonic frequency range can again be arbitrary sub-sectioned in discrete harmonic regions by maintaining synced phase-couplings to a common devisor term. In this extended BLOsc version, each frequency region can be given its own independent exponential sloping (see Figure 1).

1.1 Nyquist

A large part of the literature on the band-limitation paradigm is concerned with the problem of generating non-aliased versions of the standard oscillator waveforms found with the analog synthesizer [2][3][4]. With all Nyquist problems solved, we can safely do subtractive synthesis with our familiar palette of waveforms, but now in the digital domain. Yet, in this case, the traditional subdivision additive-versus-subtractive is far from trivial. With a subtractive scheme the developing spectrum envelope contrasts depend on the amount of filtering. With the additive BLOsc approach, large contrast can be there from the start and remain preserved when modulated. So, this earlier classification, expresses a critical division; very different musical results may emerge not only due to a difference in compositional strategy, but also due to a different valuation of the spectral factors and perceptual cues that determine the timbre of a sound.

1.2 Lower frequency limit

A peculiar perceptual phenomenon appears when the limiting frequency of the BLOsc is no longer close to the Nyquist-frequency but transposed downwards to a lower, more audible frequency setting, somewhere below 3 kHz. Typically, more or less involuntary, the BLOsc sound will attain a voice-like character, where the cutoff frequency will associate with a distinct vowel identity. A first inexplicit suggestion of an articulating voice may become more apparent, or more inevitable, when the limiting frequency or the fundamental frequency are modulated and follow distinct gestures over time. The effect is audible in S. Oishi's electronic compositions and with his BLOsc Super Collider objects you can simply explore this phenomenon yourself [6]. It is a known phenomenon. Assman and Nearey [7] already report how discrete, equal-intensity (flat) harmonic configurations, may trigger the perception of specific vowel identities and they were able to link the cutoff frequencies to intensity changes in the first formant region. Their study provides answers using a static, constant frequency viewpoint, but we were specifically interested in the dynamics.

1.3 Sharp Peak or Steep Cutoff

Formants are the designated spectrum structures that determine the vowel identity, where the first two resonances, called F1 and F2, are the most important. The formant frequencies vary with the articulated vowel, where F1 typically moves in the frequency range from 200 to 1200 Hz, while F2 can be found in the range from 500 to 2700 Hz [8]. There is an obvious overlap with the earlier-mentioned range below 3 kHz where the BLOsc attains this voice-like character, and a relevant question is: how can a steep spectrum cut-off bring up the suggestion of a dual formant resonator system?

The general notion is that the center frequencies of F1 and F2 determine the vowel identity. Although the formant peak is widely seen as the discriminating factor, this idea is not as absolute as often thought. There are different ways to conceive the spectrum level contrasts that appear below 3 kHz. In running speech formants are seldom sharp. Formants cutout distinct spectrum areas, they occupy a certain bandwidth. When formants overlap in range, they together build a raised structure, but they also build a larger level contrast seen over a wider frequency range. Even when formants are characterized as moving spectral peaks, as for instance in your cell-phone that uses an LPC-like coding technique, then still the peaks implicitly code for contrasting slopes seen over a wider frequency area.

When formants move -and they move fast in speech [9][10]- then all peaks blur while the dynamically moving edgings become the distinctive elements. Note that our ears are particularly good, or even predisposed with detecting significant differences in these fast changing spectral settings. Those who have played around with cross-synthesis (vocoding) might have experienced the following; when an arbitrary complex sound is used as a carrier and speech as the modulator, typically most of the original spectrum character of the carrier remains preserved up till the moment that a time-varying modulation brings the speech interpretation to the foreground. Note, that the dynamic aspect of our hearing is generally understood from a static frequency viewpoint. A sinusoidal sweep is modeled as a sequential rippling through successive band filters where it stops being a single coherent unit.

Plomp [8][11][15] forwarded a simple analysis system of two spectral weighting curves that performs equally well in positioning a vowel in the F1/F2 plane. The first dimension senses the specific spectrum region where F1 is most effectively varying the spectrum contrast, and the other does this for F2. It is actually the derivative of the spectrum envelope curve that does the work; the shoulders tell where the formant is and again the cut-off frequency becomes the critical factor.

It is possible to synthesize a plausible singing voice sound without modeling any formant peaks; only a flat spectrum envelop with a sharp-edged gap suffices [12]. The designated sound examples also demonstrate how a shift in the frequency location of only an up-going spectrum edge may change our perception of the voice type. Although we generally assume that our ears search for spectrum peaks, we could actually be listening for the locations of sharp spectrum edges that often coincide with the peak locations.

There is no reason to doubt the wisdom to locate spectrum peaks. What sticks out in the spectrum remains important as it informs on the precise locations of the characteristic resonance modes of a system. But this typically applies to static, time-invariant modeling. From a static viewpoint, our sectioned BLOsc approach will be cumbersome. Better or more efficient schemes can be found to precisely model a designated spectrum shape. However, many spectrum cues move with time. Moreover, spectrum characteristic can seldom be observed in full detail, as there is constant competition with features of other sounds. Catching the salient spectrum contrasts while maintaining a dynamic continuation of this contrast over both the time- and frequency-axis becomes the issue. This thinking positions our BLOsc modeling.

2. FORMULATION

In ancient times, Euclid already described in one of his "Elements" the underlying mathematical principle that the BLOsc is based on; the sum formula for a geometric series (see derivation 1).

$$z^0 + z^1 + z^2 + z^3 + \ldots + z^{N-1} = \left(z^0 + z^1 + z^2 + z^3 + \ldots + z^{N-1}\right)\frac{(1-z)}{(1-z)} = \frac{\left(z^0 + z^1 + z^2 + z^3 + \ldots + z^{N-1} - z^1 - z^2 - z^3 - \ldots - z^N\right)}{(1-z)} = \frac{\left(1-z^N\right)}{(1-z)} = \sum_{n=0}^{n=N-1} z^n \quad (1)$$

In his book: "Fractals, Chaos, Power Laws", Manfred Schroeder presents several geometrical proofs of the above derivation in what he calls "a simple case of self-similarity" [13]. Essentially, our extended BLOsc model aims at breaking up the frequency range in sections with a self-similar harmonic power development. This sectioned behavior is successively condensed using the above geometric series abstraction.

When the z in (1) is replaced by the complex exponential $z=re^{i\omega t}$, then the above sum will obtain a double identity; for the r^n multiplier it is still a "geometric series" with exponentially incrementing (or decrementing) magnitude, but the $e^{in\omega t}$ term, that unfolds into $cos(n\omega t)$ and $sin(n\omega t)$ terms, will also make it a "harmonic series" with linearly incrementing $n\omega t$ terms (2).

$$\sum_{n=0}^{n=N-1}\left(re^{i\omega t}\right)^n = \sum_{n=0}^{n=N-1} r^n e^{in\omega t} = \sum_{n=0}^{n=N-1} r^n\left(\cos(n\omega t) + i\sin(n\omega t)\right) \quad (2)$$

$$= \frac{\left(1-r^N e^{iN\omega t}\right)}{\left(1-re^{i\omega t}\right)} = q(t) = q(\omega) \quad (3)$$

The quotient in (3) that comprises the sum from (2) can be interpreted as a signal $q(t)$ resulting from an additive Fourier synthesis at time instant t, where the successive harmonic $n\omega$ terms have a magnitude that scales exponentially with the frequency index. The same quotient can also be seen to represent the result $q(\omega)$ of a Fourier analysis at frequency ω, where the analyzed signal has the form of an exponentially decaying impulse train r^n that is sampled at successive nt time intervals. So, when we listen to the periodic signal $q(t)$ synthesized by the BLOsc we actually hear the sampled circular frequency characteristic $q(\omega)$ of a truncated exponential decaying pulse train of length $N-1$ coming from a first order recursive filter (one pole). To avoid confusion, we will next stick to the time-domain signal $q(t)$ interpretation only.

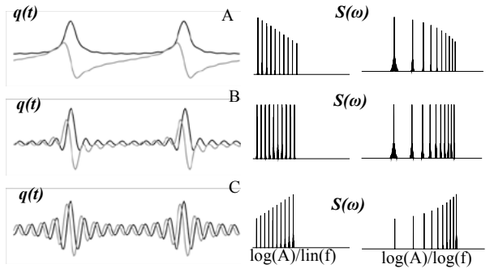


Figure 2. BLOsc signals with different r-coefficient. Signal $q(t)$ with real (all cos-phase, black curve) and imaginary (all sin-phase sum, grey curve) plus its corresponding amplitude spectrum $S(\omega)$ seen with different frequency scales. N=10 harmonics. A: down-going slope -3 dB/harmonic ($r=2^{-½}$), B: flat envelope (r=1), C: up-going +3 dB/harmonic ($r=2^{½}$).

3. REALIZATION

Our BLOsc will always generate a so-called: "analytical signal"; two outputs with a 90-degrees phase difference. Except for a DC-offset, the real (cosine-sum) and imaginary (sine-sum) output will sound identical, as our ears will be deaf to this constant within-period phase difference [15]. Maintaining complex calculus all-over serves several purposes: (I) rounding errors will stay low, (II) the processing scheme and dynamic control stays simple as you remain close to the formulation which also makes it easier to later section the mechanism and (III), spectral blocks do not mirror on zero frequency, which is essential for a later auto-convolution by raising time domain power. Even if the calculation load is considered, there will be no benefit in converting to an all real-valued signal implementation. The i86-based processors found in many computers have inbuilt CORDIC-based [14], double-argument instructions that do a *polar-to-Cartesian* or *Cartesian-to-polar* conversion as fast as a floating-point multiplication or division. The *exp()* and *log()* function are comparably fast. Any "vintage" digital version that builds on wavetables will not have a different sound; it will only be slower, less flexible and stand in the way of a further development of the BLOsc scheme in a new musical direction.

3.1 Decay control

In computer sound synthesis environments like Chuck! or Super Collider (SC), standard implementations of the BLOsc can be found. Typically, only the spectrally flat ($r=1$) option is advertised. This simplification in a way downgrades this powerful oscillator mechanism, as it almost unnoticeably directs users in the direction of a subtractive synthesis paradigm.

The idea of a dynamic r-modulation was already there in the original description by Moorer [1]. The r-coefficient allows a fluent and precise control of the spectrum slope ranging from a pure sine (the fundamental) to a flat spectrum of a band limited impulse train (BLIT), and even beyond to a configuration where the spectrum gets progressively weaker at the low frequency-end to the point that only the highest bounding-harmonic resides (Figure 2).

4. EXTENSIONS

4.1 Sectioning

When frequency (and phase) maintain their unidirectional interpretation, this means that complete self-similar harmonic progressions can be shifted, or rotated, without their content further spreading over the frequency axis. Such a harmonic section will preserve the sine-in-sine-out linear system property of all its constituting components. The instantaneous frequency and amplitude can be modulated or swept as with one single sine, with the additional modulation of the slope coefficient r as an extra bonus.

Building on this line of thought, an additive synthesis scheme is implemented, where instead of single harmonics, complete harmonic sections of variable width and decay are controlled separately, but where all sections are still sync to the same fundamental frequency base (see figure 3). For each section only the first harmonic components needs to be generated; the first harmonic in the next section is the "N" of the previous section.

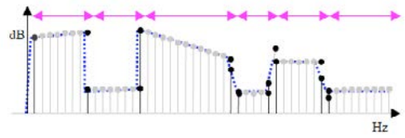


Figure 3. Coupled band-limited oscillator sections. This sketch was used to generate the signal in Figure 1.

Note that section levels may jump at each bound; a finite slope value can be modeled by introducing a new (close) section bound. With each section another quotient $q(t)$ is associated. If all sections share the same r-coefficient as an overall spectrum slope parameter, then all quotients may share the same denominator base. In its complex-logarithmic representation, the whole scheme turns into a simple series of additions and subtractions within a dB/Hz scaled framework.

As all frequency multiples stem from the same phasor, we are free to choose any harmonic block-width within this additive scheme and do overlap-add in the frequency domain. There is no need for an equal center spacing or constant overlap factor.

4.2 Modulation

The amplitude (magnitude of $|q(t)|$), the fundamental frequency ω, and the slope coefficient r, can all be instantly modulated without violating the generalization. The spec-

tral spreading can be predicted by extrapolating the modulation rules for a single frequency component. However, any instantaneous update of the bounding harmonic number (the ***N***), will generally result in a glitch. At the crossing of the zero-phase instant, the rate of change in the phase curve and log amplitude curve are minimal. At this point in the fundamental period all sine sums are crossing zero and all cosine sums reach their maximum. This is the normalization point where amplitude sums in time and frequency domain equate. It is thus the best point to switch over N, and/or change signal power number (discussed later). To dynamically change the "N" the BLOsc must operate in a period-by-period mode. The zero-phase instant is also the least-penalty point to start-stop the oscillator in a one-shot period mode. Although the abrupt begin and end violate the infinite time principle the BLOsc is based on, that does not mean that the sound results will be less interesting.

4.3 Alternative envelope controls

In the GENDY model from Xenakis, waveform breakpoints are varied using a stochastic model. This idea can in a direct way be ported to a frequency domain sectioned BLOsc implementation, where the breakpoints vary on a dB/Hz scales, while updating is done in a period-by-period fashion. As a different overlap is allowed, the breakpoints may also be randomly redistributing on a logarithmic frequency scale to better agree with a perceptual (critical band) organization. A log(A)/Log(f) setting compares to the domain where the MFCC's are defined from. Thus the sectioned BLOsc presents a simple scheme to re-synthesize this spectrum envelope abstraction using an arbitrary fundamental frequency carrier.

5. CONVOLUTION AND CHANCE

5.1 Raising power in the time domain

Fourier theory presents the principle that multiplication in the frequency domain compares to a convolution process in the time domain, but the opposite is also true. So, when the analytical signal ***q(t)*** is multiplied with itself, producing the (still complex-valued) squared signal $q(t)^2$, this simple operation will correspond to an auto-convolution (or cross-correlation, or filtering) in the frequency domain. The principle is demonstrated in Figure 4, where the harmonic series *1..6,* is convolved with itself by simply squaring ***q(t)***. This brings about a new harmonic series with the information spread over double the frequency width, but with still the original harmonic spacing. Rising ***q(t)*** to the third power will spread the information up to harmonic 18. For signal ***q(t)*** the spectrum envelope is flat (constant, zero order). With the squared signal $q(t)^2$ the spectrum envelope goes linearly up/down (first order), and with a cubic signal $q(t)^3$, the spectrum follows a parabolic (second order) curvature. A comparable generic scheme that builds from a flat zero-order kernel, to higher order shapes is seen with B-spline interpolation [16].

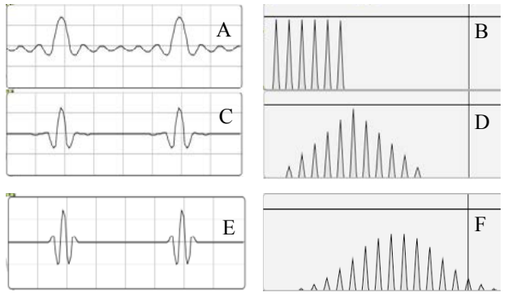


Figure 4. Raising power in the time domain. Signal ***q(t)*** (A) real (cosine-sum) only, for BLOsc (*r=1* & *N=6*), $q(t)^2$ (C) and $q(t)^3$ (E). Corresponding *Lin(A)/Lin(f)* spectra (B, D, F) demonstrate the convolution effect in the frequency domain.

In the example in Figure 4, deliberately six harmonics of equal amplitude were chosen to draw a parallel to the uniform chance function that is seen for each digit of a six-sided dice. When more dice are thrown at the same time, each dice will be an identical independent distributing (IID) process. As the chance functions combine as in a convolution procedure, the harmonic distributions seen for $q(t)^2$ and for $q(t)^3$ will match to the probability functions that result when two, and when three dice are thrown together. Unfortunately, the above spreading mechanism will disassociate when fractional powers are used. This is a pity as it could have presented us a simple direct scheme to arrive at a linear spectrum slope control in a log(A)/Log(f) scale setting.

5.2 Formant shaping

To exploit the above spreading mechanism at least two harmonics are needed. An initial setting with equal harmonic amplitude compares to the 50% chance seen with a coin flipping process. Following this analogy, we can predict that each power increment will successively add a new harmonic to the series, where the numbers from Pascal's triangle will predict the amplitudes series. For not even that high (third) power the harmonic amplitude distribution will already reasonably approximate a Gaussian shape, as shown in Figure 4F.

Note that the frequency information will still be bandlimited to a width set by the power-index for the signal in the time domain. On the logarithmic dB-scale the approximated Gaussian shape reveals its minus-squared dependency as an inverted parabola (see Figure 5d). The thus produced spectral prominence can be used to model a formant resonance in an additive synthesis scheme. It follows the thinking that is also seen with the VOSIM, FOF and PAF models [17][18][19].

A disadvantage of this time-domain power-controlled formant-shaping approach is that the bandwidth of the peak will increase with the fundamental frequency. It can thus be difficult to model real sharp peaks, as to give formants steep shoulders; high signal-powers are needed, which also widens the distribution. This problem can be partly solved by varying the ***r***-coefficient of the initial pair (see Figure 5).

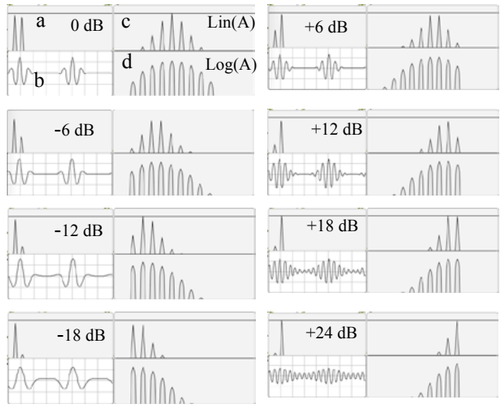


Figure 5. Center shift as function of ***r.*** In each panel: (a) initial harmonic balance, (b) signal $q(t)^8$, (c) resulting spectrum as Lin(A)/Lin(f) and (b) as Log(A)/Lin(f).

The resulting distribution will reshape from skewed-to-the-left, via normal, to skewed-to-the-right bound due to a gradual rebalancing of the prominence within the initial harmonic pair. Such a spectral progression will mimic in a realistic way a moving formant, however, with sudden stops at the region bounds.

6. CONCLUSION

One chooses this extended BLOsc model for the strict frequency partitioning that it holds, even on modulation. Complete harmonic sections, or formant like structures, can be moved as one unit while preserving the same degree of spectrum contrast. It is thus possible to stay sharp in frequency while following a sharp defined path in time. Time-invariance is a major constraint in filter design. To realize steep spectrum contrasts we generally need higher-order (inflexible) filter-structures with long impulse responses. For this reason, this additive scheme can offer you "dynamic spectral consistencies" that are hard to realize with any subtractive (filtering) synthesis scheme, and perhaps new and interesting sound results.

7. REFERENCES

[1] J.A. Moorer, "The synthesis of complex audio spectra by means of discrete summation," *J. Audio Eng. Soc.,* 24, pp. 717–727, 1976.

[2] V. Välimäki, and A. Huovilainen. "Oscillator and filter algorithms for virtual analog synthesis." *Computer Music Journal* 30.2 (2006): 19-31.

[3] V. Välimäki, J. Pekonen, and J. Nam. "Perceptually informed synthesis of bandlimited classical waveforms using integrated polynomial interpolation." *The Journal of the Acoustical Society of America* 131.1 (2012): 974-986.

[4] J. Nam, et al. "Alias-free virtual analog oscillators using a feedback delay loop." *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx09)*, Como, Italy, September 1–4. 2009.

[5] J. Pekonen, et al. "Variable fractional delay filters in bandlimited oscillator algorithms for music synthesis." *Green Circuits and Systems (ICGCS), 2010 International Conference on. IEEE*, 2010.

[6] S. Oishi, "Timbral Movements in Electronic Music Composition". Master thesis, Institute of Sonology, Royal Conservatoire, the Hague, 2015.

[7] P. F. Assmann, and T.M. Nearey, "Perception of front vowels: The role of harmonics in the first formant region." *The journal of the Acoustical society of America*, 81(2), (1987): 520-534.

[8] L.C.W. Pols, H.R.C. Tromp, and R. Plomp. "Frequency analysis of Dutch vowels from 50 male speakers." *The journal of the Acoustical Society of America* 53.4 (1973): 1093-1101.

[9] T. Gay, "Effect of speaking rate on diphthong formant movements." *The Journal of the Acoustical Society of America* 44.6 (1968): 1570-1573.

[10] D. J. Hermes. "Vowel-onset detection." *The Journal of the Acoustical Society of America* 87.2 (1990): 866-873.

[11] L. Pols, CW, LJ Th Van der Kamp, and R. Plomp. "Perceptual and physical space of vowel sounds." *The Journal of the Acoustical Society of America* 46.2B (1969): 458-467.

[12] http://kc.koncon.nl/staff/pabon/SingingVoiceSynthesis/CutoffF3/CutoffFreqF3.htm, retrieved Feb 29, 2016.

[13] M. Schroeder, "Fractals, Chaos, Power Laws," W.H Freeman and Company, New York, 1991.

[14] O. Spaniol, "Computer Arthmetic, Logic and Design," John Wiley & Sons, New York, 1981.

[15] R. Plomp, Aspects of tone sensation: A psychophysical study. Academic Press, 1976.

[16] Unser, M. Splines; A Perfect Fit for Image and Signal Processing. *IEEE Signal Processing magazine*, 16. (1999) 6.

[17] M. Puckette, "Formant-based audio synthesis using nonlinear distortion." *Journal of the Audio Engineering Society* 43.1/2 (1995): 40-47.

[18] Tempelaars, Stan. "The VOSIM signal spectrum." *Journal of New Music Research (AKA Interface)* 6.2 (1977): 81-96.

[19] X. Rodet, Y. Potard, and J.B. Barriere, "The CHANT project: from the synthesis of the singing voice to synthesis in general." *Computer Music Journal* 8/3, (1984) pp. 15-31.

Granular Spatialisation, a new method for sound diffusion in high-density arrays of speakers and its application at the Spatial Audio Workshops residency at Virginia Tech (August 2015) for the composition of the acousmatic piece *Spatial Grains - Soundscape No 1,* for 138 speakers

Javier Alejandro Garavaglia
Associate Professor
London Metropolitan University, UK
jag@jagbh.demon.co.uk

ABSTRACT

This paper was originally submitted for the ICMC 2016 together with the acousmatic piece "Spatial Grains, Soundscape No "1, describing in theory and practice the usage of Granular Spatialisation. Granular Spatialisation is a new and particular case of an on-going development of diverse systems for automatic, adjustable and time-dynamic spatialisation of sound in real time for high-density speaker arrays, and can be therefore contextualised as a further and special case of development in the practice-based research of the author of this paper about the main topic of full automation of live-electronics processes, as explained in [1] and [2]. The paper considers both the theoretical background and the initial phases of practice-based research and experimentation with prototypes programmed to diffuse sound using spatialised granulation. The second part of the paper refers to a recent experience during a residency at the Cube, Virginia Tech, using Granular Spatialisation within an array of 134 + 4 loudspeakers for the diffusion of the acousmatic composition jointly submitted herewith. Seeing that in the past 40 years, the number of speakers for the diffusion of acousmatic music has constantly increased, this paper finds pertinent the main question of this ICMC: "Is the sky the limit?" with regard to the number of loudspeakers that can be used in acousmatic sound diffusion.

1. INTRODUCTION

Based on the general concepts of granular synthesis developed by Truax [3] and Roads [4], which are respectively predicated on Gabor [5] and Xenakis [6], Granular Spatialisation (GS hereafter) transfers the common parameters of a grain (such as grain time, window/envelope type, inter-grain time and grain overlapping time) to the movement *in real time* among loudspeakers within a multichannel environment.

Although GS works appropriately from a 4.0 surround system onwards, the ideal speaker configurations are those with a large number of loudspeakers located in different areas within a performance space, which can be, for example, a concert hall or a gallery space, the latter mainly for sonic or audiovisual installations.

2. GRANULAR SPATIALISATION: MAIN FEATURES

GS has been conceived to work in both performance and/or installation environments, with at least a 4.0 (quadrophonic) sound output. The directionality of the sound can be in either surround or in any other type of directional arrangement. However, the main goal is to work in loudspeaker configurations, which are significantly larger, for example, the *Klangdom* at ZKM (*Zentrum fur Kunst und Medientechnologie*, Karlsruhe, Germany), which consists of an array of 47 speakers, or the CUBE at Virginia Tech (US) with 147. Further similar venues are discussed in the conclusion (section 4) of this paper, with regard to both their own characteristic settings and to how to approach the diffusion of acousmatic music in each case.

A grain can be defined as a small particle of sound, typically of a duration between 10 to 50 milliseconds[1] consisting of two main elements: a signal (which can be produced either synthetically or from an already recorded sound) and an envelope, which shapes the signal's amplitude. Gabor [5] called these small particles of sound "a sound quantum" because, when too small, such particles cannot be perceived by our hearing as sound.

In sound synthesis, grains are normally used in big amounts per second, producing rich sound results resembling clouds made of those grains, where the perception of each grain is fully lost, but the effect of grains acting together is not. From the visual point of view, paintings from the pointillist period (mostly those by Paul Signac and Georges Seurat) are a good visual analogy to granular synthesis clouds: from a close perspective, pointillist forms can be seen as exclusively made from tiny little coloured dots (similarly to digital pixels); on the other hand, from a farther perspective, forms from those colourful dots are fully revealed. In granular synthesis, the more dense the cloud, the richer the harmonic texture of each sound moment will be. Roads [4,7] classifies sound granulation in different types, mainly the difference between *synchronous* and *asynchronous* granular synthesis. Synchronous granular synthesis works with grains that are all separated by whether the same amount of time, or at least by some type of linear relationship. On the other hand, asynchronous granular synthesis does not present such a strict linear relationship as the synchronous type, and therefore the relationship will typically contain random elements with no linear common elements amongst them as a consequence.

The goal of GS is to translate all of these main features of granular synthesis into the spatial domain *in real time* within a high-density array of loudspeakers. This allows for the most important contribution of GS: its capability to produce flexible times between 1 millisecond to any longer duration among each speaker in the array. The prototypes developed so far utilise mostly *synchronous spatial grains* including clouds, the latter mostly through the usage of diverse and several octophonic settings within the high-density array of loudspeakers. Most prototypes utilise a constant grain duration per loudspeaker, defined by a constant frequency input, which provides both the time that elapses across the entire array of loudspeakers as well as the duration of each spatial grain for each loudspeaker. Hence, and regardless of the actual number of loudspeakers in the array, the principle in which prototypes work is basically the same. Figure 1 below shows single synchronous spatial grains in a 4.0 array. There are however some prototypes which either increase or decrease the grain time between loudspeakers, constantly changing the duration of grains with the entire array of speakers. These are nevertheless synchronous spatial grains, following the definition by Roads already mentioned above [4,7].

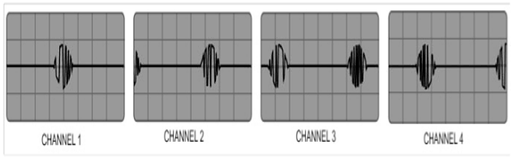


Figure 1. Synchronous spatial grains in a 4.0 array.

Although grains produced by granular synthesis DSP can be indeed diffused in a multichannel environment, the concept of GS proposes to produce the grains in real time, therefore, at the very moment in which the movement between speakers occurs and not before, as the grains are solely the result of the sound diffusion within the loudspeakers array and not synthesised. For GS to happen, a constant signal flow – either a constant, regular signal (such as white noise or a sine wave), or a concrete recorded sound – is required to spatially granulate its output within the multichannel environment. Although spatial grains can be of any size, those specially effective for GS diffusion with a granular aural effect are those which take less than 100ms to travel between any two loudspeakers within any multichannel environment. Moreover, the number of channels of a diffusion array plays a vital role in how the spatial granulation will be not only produced but also programmed, as a large array of loudspeakers within a space can transport the grains across that space much more efficiently and clearly rather than using reduced multi-track systems (such as, for example, a quadrophonic array). Hence, distance can help with the aural recognition of the positioning of the grains, mostly in those cases of spatialisation of very short grains. Figure 2 below shows in this case, that the window for the grain will be 200 milliseconds (for a rotation frequency of 5 Hz), which is the amount of time in which the full four-channel array will be used for the granulation, whilst the spatial grain itself is a quarter of that, in this case 50 milliseconds.

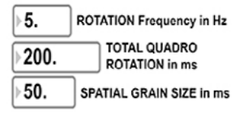


Figure 2. Calculation included in the granular spatialisation software for a quadraphonic granular diffusion. Four spatial grains of 50 ms each will be produced in each of the speakers for a full coverage of the array.

Figure 3. Envelope for grain shaping in a 4.0 surround system, in this case a Gaussian envelope covering only a ¼ of the entire duration of the rotation time window.

Figure 3 above shows how a Gaussian envelope (bell-like shaped envelope, very useful in granular synthesis, see more information later in this section) can be applied to those figures shown in Figure 1 for a 4.0 surround sound system array: the duration of the grain envelope is only a ¼ of the length of the entire window, whilst ¾ of it is silence (which is the time needed for the other three channels to produce the grains). The envelope is thereafter delayed for each speaker by the duration of each grain.

3. GRANULAR SPATIALISATION: CHARACTERISTICS

3.1 Main Characteristics

The diffusion prototypes of the systems programmed at this stage share the following main characteristics:

1. all prototypes have been programmed using the Max 6 software package;
2. The granulation of sound occurs at the specific moment of sound diffusion. Hence, each spatial grain is produced at the precise moment of sound diffusion within a high-density loudspeaker array, not before. Spatial

[1] The duration of grains can vary from case to case, and therefore, durations of 1 to 100 milliseconds can be also considered for this purpose.

grains are therefore in charge of the sound's diffusion within the speaker array: the granulation and its aural effects happen in real time at the very moment of spatialisation;

3. One grain per loudspeaker diffusion: the fundamental concept behind GS is the usage of ideally one spatial grain per loudspeaker (which could be even less than one grain, in the special cases of overlapping grains between speakers, explained later in this paper), by exploiting the physical characteristics of grains for each of the speakers within the array, instead of simulating virtual locations as it is the case, for example, with the usage of ambisonics. Hence, each spatial grain has its short specific temporary location in the multichannel system at any given time. This offers a different direction and conception compared to existent development and research in the area, as for example, Scott Wilson's Spatial Swarm Granulation [8], which presents an implementation for dynamic two or three dimensional spatial distribution of granulated sound (therefore, granular synthesis produced in advance of its spatialisation) over an arbitrary loudspeaker system;

4. Granulation effect created within the diffusion: apart from diffusing sound within an *n* number of loudspeakers, GS produces aurally a granulating effect through the diffusion, which varies in its intensity depending mainly either on the type of grain shape or on the overlapping grains between two contiguous loudspeakers or on both characteristics applied together. The diffusion can also comprise clouds of grains, depending on the density of the granulation applied;

5. Different types of spatial grains: although the aim GS is to use any type of grains for the granular spatialisation of sound, so far only synchronous spatial grains were used, mostly due to either their regularity or their linear relationship, which provides for a clear tracing for the signals diffused. Diffusion with exclusively asynchronous spatial grains is also envisaged to be used for either creating dense clouds of spatial grains or for the usage of random duration for the grains. However, at present, both cases need still proper programming and experimentation;

6. Grain-time control between each speaker in the array: the system is conceived to work by controlling the duration (and therefore, the length) of the resulting spatial grains. Although the spatial grain-time could be of any length, for the special case of spatialisation *through* grains, an ideal duration between two speakers would be between ca. 10 ms and 100 ms, although the system uses also both shorter and longer times, the latter in the case of overlapped grains among contiguous loudspeakers. The duration of the grains is defined in this system – as shown in Fig. 2 above – by two parameters:

(a) a rotation frequency (programmed in Hertz), which establishes the time for the spatial grains to cover the entire array of speakers, as defined for each specific case;

(b) the actual number of speakers within the array.

Hence, by a fixed rotation frequency, the higher the number of loudspeakers included in the array, the shorter spatial grains will become. Also, the shorter the grain, the more the typical s*patial granulation* effect can be perceived, including its typical granulated noise;

7. As sounds are constantly and only diffused via spatial grains, the localisation of diverse *spectromorphological*[2] aspects of these sounds – according to Smalley's concept about soundshapes [9] – constantly vary their position in the high-density-speaker array, creating a rather rich and varied s*patiomorphology*, which, also according to Smalley, defines the exploring of spatial properties and spatial changes of sound(s) [9,10];

8. Grain envelopes: spatial grains work in this development with different typical smooth table functions for diverse window envelopes shapes, such as the Gaussian and Quassi Gaussian (Tukey) types, but – depending on the type of granulation desired – they can include sharper envelopes such as triangular, rectangular, etc.;

9. Directionality of the diffusion: spatialisation occurs so far with either a clockwise or an anti-clockwise movement of sound within the arrays of a selected number of loudspeakers. Both directions can use either synchronous or asynchronous spatial grains. As mentioned above, random diffusion and asynchronous spatial grains have yet to be implemented within this development;

10. Grain duration: spatial grains in the prototypes developed and tested so far have either a fixed duration or can also dynamically increase or decrease their duration whilst travelling across an array of multiple speakers. In the latter case, grains either accelerate or slow down the circulation of the granular diffusion. This is a smooth and gradual usage of synchronous spatial grains, as they still possess a linear relationship among them;

11. Flexible speaker array constellation: GS can be applied to any set of multi-speaker diffusion system, from 4.0 to an *n* number of loudspeakers in either multiple arrays of, for example octophonic clusters, or using the entire array at disposal. This allows the system to diffuse sound with a typical spatially designed grain characteristic in particular environments. However, the best arrays are those with around 100 or more loudspeakers, as it is explained later in this article;

12. Overlapping of spatial grains (soothing the granulating effect): although GS has been conceived to produce a granulating effect in the overall outcome, thus, allowing for grains diffusing each particular sound per loudspeaker to be heard as such as well as the sound they transport or diffuse, the concept can also be used to diffuse sound more smoothly by overlapping grains amongst two, three or four loudspeakers. Therefore, the question of overlapping or not spatial grains is relevant with regard to how the effect of granulation should be perceived in the spatialiasation. In its pure, original conception, only one full grain per loudspeaker should be heard, meaning that

[2] Spectromorphology is the perceived sonic footprint of a sound spectrum as it manifests in time [9].

all of the other channels are muted. Through the start and end of the grain envelopes, the effect hereby adds a desired granulation noise the higher the rotation frequency is increased. However, overlapping grains between contiguous loudspeakers soothes the process. In order for spatial grains to be still perceived as such, overlapping neither should be massively long nor should it happen across multiple speakers. Hence, only 2x, 3x and up to a maximum of 5x overlapping should be used hereby, with rather short grain durations, in order for the granulation effect to be still perceived, albeit soother than without overlapping.

3.2 Composing & programming GS for the diffusion of sound at the CUBE's high density speaker's array at Virginia Tech (US).

This section briefly describes the experience of composing and diffusing in concert the acousmatic piece *Spatial Grains, Soundscape No 1* inside the Cube at Virginia Tech in August 2015 during a short residency, mixing together sounds from several and very different parts of the world. The spatial diffusion of this imaginary soundscape takes place exclusively through the usage of GS.

The Cube space has an array of 138 loudspeakers, (a figure that includes 4x subwoofers), all of which are distributed between three floors: ground (64x + 10x), two Catwalks and the grid layer (the latter three, with 20x speakers each, whereas the first Catwalk includes the 4x subwoofers)[3]. Although it is not a huge hall, and moreover and in spite of its name, it is of slightly rectangular form, distances in the plane (for plane waves) are relatively short. However, after experimenting on a daily basis with the system during the residency, distance could be indeed perceived in its 3D constellation and therefore, sounds from either the grid, or from each of the Catwalks were very much identifiable with regard to their location in the space and most surprisingly, even from the four subwoofers in the first Catwalk. In order not to use the entire array of the Cube for each sound – which would have been against both the spectromorphological and spatiomorphological characteristics of those sounds in most of the cases – reduced and located sub-arrays of loudspeakers within the Cube were programmed for the composition. The majority of these sub-arrays are octophonic, with some exceptions, such as a 10x speaker sub-array (stage speakers) in the ground floor and further two sub-arrays with twenty-four loudspeakers each in the first and second floor. This conception allowed for a much more creative and sensible manner of propagating sound, and at the same time, for some sounds to be located at only a restricted area of the entire speaker array, due to, for example, their spectromorphological and spatiomorphological characteristics. The main idea of the piece was to use many different types of sounds, each of which required a player similar to the ones described in figures 5 and 6 below. Each player contains the definition of the type of

[3] There is actually a total of 147 speakers in the Cube space, thus 9 speakers more than herewith described, but these (9x Holosonic AS-24) were not available during the residency.

window for the grain envelope, the direction of the spatial granulation, the rotation frequency – that is, the duration for the grains to complete the n-number of speakers of the sub-array cycle – and the grain size, determined by the division between the rotation frequency and the number of channels within the sub-array.

The players for the spatial granulated diffusion of the piece were included in two main and separated patchers programmed in Max, both of which considered different aspects and possibilities of usage of the Cube's loudspeaker array system. The first Max patcher is based on a mixture of several players, each of which plays only mono files with an output of 8x channels, with the exception of one single player designed for quadrophonic output (for 4x subwoofers in the first floor, front, rear right and left sides), one extra player for the 10x stage loudspeakers placed in a surround disposition on the floor (JBL LSR6328P speakers, different to the other 124), and 2x players using 24x channels each. This first Max patcher did not repeat any loudspeaker in any of the 8x, 10x or 24x settings, as it is shown in figure 4.

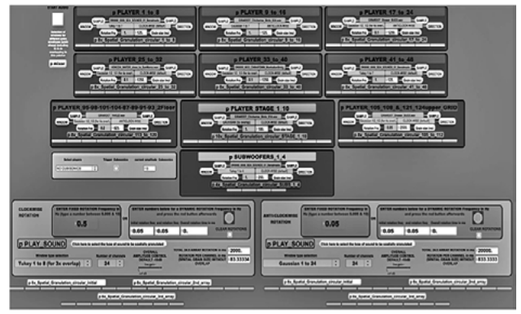


Figure 4. First Max patcher with 4x, 8x, 10x and 24x sub-arrays of speakers for the acousmatic composition *Spatial Grains, Soundscape No 1*.

Figure 5 shows one of the two 24x speakers arrays situated between the upper two Catwalks, with a rather elliptic distribution of speakers, ideal to spatialise sounds of sources such as birds, insects, etc.

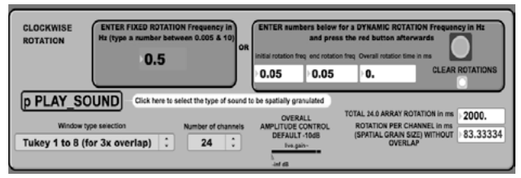


Figure 5. One of the two players of 24x speakers in the first Max patcher. This player can either have a fixed rotation frequency or it can vary constantly the speed of spatialisation, therefore continuously changing the grain size.

Figure 6 below shows *Player_Stage 1 to 10* (the 10x stage speakers on the ground floor), in which the player diffuses the spatial grains within a 10.0 surround sub-array. In spite of this example, most of the players in the first Max patcher have an octophonic surround disposition within each of the floors. The directionality of the granu-

lated diffusion can be either clockwise or anti-clockwise for any of the players.

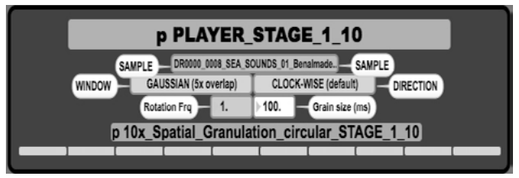


Figure 6. Stage 1-10 player.

The second of the Max patchers features a continuously linear diffusion on all 124 speakers within the entire Cube excluding the 10 stage speakers. Both Max patchers allowed for the usage of the full available array of 138 speakers inside the Cube space.

There was a substantial difference in the disposition of outputs for each of the two Max patchers, with the clear intention of creating different virtual spaces within this large array of speakers, in order for different aspects and elements of the sounds included in the soundscape to become clearly identifiable within the space. With the exception of both 24x spatial grain players as described in figure 5, all of the other players have a surround disposition in each of the three floors with a unique selection of loudspeakers for each player with regard to their exact position within the room. As an example, figure 7 below shows the octophonic surround disposition within the 64 speakers of the ground floor of the Cube of one of the octophonic players. The 8x surround sub-array follows the pattern for which each speaker within the sub-array is equally separated by every seven contiguous speakers, in order to create an individual location for each of the 8x sub-arrays around the audience. Hence, there are 6 octophonic players in the first Max patcher, all of which have a different configuration with regard to their actual speaker numbering (within the Cube, they are numbered from one to 64), whereas no speaker was repeated for any of the players/sub-arrays. The lack of speaker repetition, plus the different settings of spatial grain duration and diffusion direction (clockwise or anti-clockwise), created a fine thread of layers of different sounds and their diverse movements within that particular floor.

The disposition shown in figure 7 below is for *player 33 to 40* only. As mentioned above, all of the other 8.0 players in the ground floor utilise similar combinations and separations for the loudspeakers, without repeating any of the speakers in any of the different combinations. The main reason for such an octophonic surround disposition of speakers in this Max patcher is to avoid that a sound is diffused using only eight contiguous speakers, which would confine that sound to a strict particular and small area of the entire array within each of the floors within the Cube.

With regard to this composition and its performance during the presentation of this paper at ICMC 2016, the main challenge relied on adapting the spatialisation of the composition for a much smaller set of only 16.2 for demonstrations (the submission was originally made to use the 192 speakers of the *Game of Life* system, but the system was not made available for the ICMC in the end).

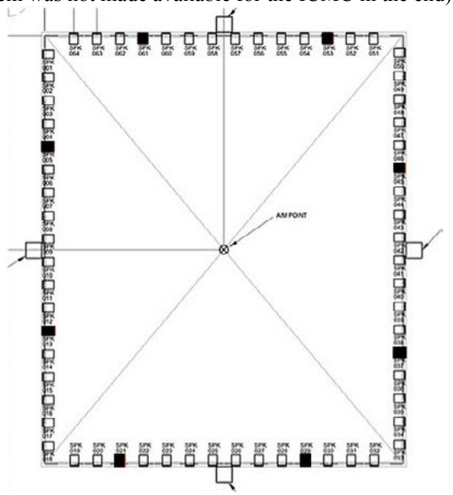


Figure 7. Layout of the ground floor of the Cube space with 64 loudspeakers. The speakers filled in black are those used in one of the octophonic players. The four bigger squares represent the 4x subwoofers.

4. CONCLUSION

The experience in the Cube has shown, that GS can be applied in full for an independent and new manner of diffusing electroacoustic music in high-density arrays of loudspeakers, in spite of the fact, that it can still be effectively used in small diffusion systems such as 4.0, 5.0 or 8.0.

One of the main conclusions after using for the first time GS in a proper high-density array of speakers, is that the Max patchers programmed for the performance of the composition can only be used in the Cube, whilst their translation to any other space and system, such as, for example, the *Game of Life* system in Utrecht (192 speakers), or the BEAST in Birmingham, UK (which counts ca. 100 different loudspeakers) or the *Klangdom* at ZKM, Germany (47 speakers) will fully change the manner in which the piece is performed, and therefore, presents a major challenge with regard to the treatment of spectromorphological and spatiomorphological characteristics of the sounds and their spatialisation within a completely different arranged and conceived environment. From the above, only the Cube's and the *Game of Life's* systems share a basic common WFS[4] conception though.

One of the most relevant differences amongst those spaces mentioned above is related to how timbre works. The Cube is clearly conceived for WFS (in 3D), supported by its equal type of 124 loudspeakers in all three Catwalks and the grid. On the other hand, the BEAST in Birmingham, UK, is assembled with several different types of loudspeakers instead of one or two types, featuring as a consequence diverse frequency responses (the most notable, the arrays of tweeters on the roof, to enhance just high frequency sections of the spectrum), and therefore, diffusion of the same piece with this system would present fundamental changes in timbre compared to the Cube's configuration. This implies that, with regard to spatialising the same acousmatic piece in differently designed spaces, several issues must be considered. On the one hand, the programming of the diffusion of pieces must be radically adapted, in order to suit at its best the characteristics of the system used (e.g. the type and number of loudspeakers, their disposition within the space and the variety of applying – or not – different types of loudspeakers inserted within a complex multi-channel array). On the other hand, due to the fact that dramaturgical and timbrical aspects of the pieces can drastically vary from place to place and system to system, the tactics involved in how to use a particular multi-speaker system must be constantly reconsidered, in order to obtain a sound diffusion that suits both the composition and the space with regard to the composition's spatiomorphological contents and potential. This is due to the fact that he usage of different types of loudspeakers in each space may or may not suit the quality of the diffused sounds (including their timbre) and therefore, may or may not serve the intended dramaturgical effect of the composition. With special regard to timbre, GS performed rather equally at every section of the Cube and it proved excellent for carrying sound through the three catwalks and grid with a rather equal timbrical characteristics (most noticeable in those speakers with a WFS configuration). Empirical tests to analyse GS's timbrical response were not possible during the short residency at the Cube, but are planned for the future, in order to fine-tune the system.

From all the above, it is clear that GS is suitable to be adapted for any of those systems already mentioned in this paper as well as other, similar ones, with particular results expected from each different system and space.

As mentioned in the abstract, the number of speakers in the last 20 years seems to only increase: at the Cube in Virginia Tech there are 148 speakers, at IRCAM there are 246, the *Game of Life* has 192. Thus the pertinent question: is the sky the limit for the number of speakers in the future? The ICMC 2016 is asking this question as a conference theme, which seems more than appropriate hereby. With regard to GS, it is clear that it is a new option, which can deal with any number of loudspeakers in order to spatialise sound within any space. In spite of its first full prototypes (as those used at Virginia Tech) were programmed with Max, there is nothing against the inclusion of any other software to do this, such as, for example, Supercollider. In fact, this is my plan for the future of this research. However, Max has proven so far to be a reliable tool for this first and absolved part of this on-going practice-lead research. Thus, depending on the architecture, acoustics, and electroacoustic system design (i.e. the number of loudspeakers in the high-density array) where GS can be applied, new concepts and developments in sound spatialisation and the diffusion of acousmatic music can be explored and expanded.

[4] WFS (Wave Field Synthesis) is a spatial sound field reproduction technique performed via a large number of loudspeakers, in order to create a virtual auditory scene over an also large listening area. The concept was initially formulated by Berkhout [11, 12] at the Delft University of Technology at the end of the 80s, based also on previous research, homophony

5. REFERENCES

[1] J. Garavaglia, "Raising Awareness About Complete Automation of Live-Electronics: a Historical Perspective," *Auditory Display*, *6th International Symposium CMMR/ICAD 2009, Copenhagen, Denmark, May 2009. Revised papers - LNCS 5054*. Springer Verlag. Berlin, Heilderberg, pp. 438–465, 2010.

[2] J. Garavaglia, "Full automation of real-time processes in interactive compositions: two related examples," *Proceedings of the Sound and Music Computing Conference 2013* edited by Roberto Bresin, KTH Royal Institute of Technology, Stockholm, 2013, pp. 164–171, 2013

[3] B. Truax, "Real-Time Granular Synthesis with a Digital Signal Processor," *Computer Music Journal* 12(2), pp. 14–26, 1988.

[4] C. Roads, et al. (ed), *The Computer Music Tutorial*. Cambridge: MIT Press, p. 175, 1996.

[5] D. Gabor, Dennis. "Acoustical Quanta and the Theory of Hearing." *Nature 159* (no. 4044), pp. 591-94, 1947.

[6] I. Xenakis, *Formalized Music*. Bloomington: Indiana University Press, 1971.

[7] Roads, C. Microsound. Cambridge, MA: The MIT Press, p. 88, 2001.

[8] Wilson, S. 2008. "Spatial swarm granulation". ICMC 2008 Proceedings, hosted by Michigan Publishing, a division of the University of Michigan Library, http://quod.lib.umich.edu/i/icmc/bbp2372.2008.127/1.

[9] Smalley, D. 1997. "Spectromorphology: explaining soundshapes". Organised Sound 2(2), Cambridge University Press (UK), pp. 107–26.

[10] Smalley, D. 2007. "Space-form and the acousmatic image". Organised Sound 12(1), Cambridge University Press (UK), pp. 35–58. doi: 10.1017/S1355771807001665

[11] Berkhout. A. J., "A holographic approach to acoustic control". J*ournal of the Audio Engineering Society*, 36, pp. 977–995, 1988.

[12] Berkhout. A. J., D. de Vries, and P. Vogel. "Acoustic control by wave field synthesis". J*ournal of the Acoustic Society of America*, 93(5), pp. 2764–2778, 1993.

Motivic Through-Composition Applied to a Network of Intelligent Agents

Juan Carlos Vasquez
Department of Media, Aalto University
juan.vasquezgomez@aalto.fi

Koray Tahiroğlu
Department of Media, Aalto University
koray.tahiroglu@aalto.fi

Johan Kildal
IK4-TEKNIKER
johan.kildal@tekniker.es

ABSTRACT

This paper presents the creation of two music pieces using motivic through-composition techniques applied to a previously presented NIME: a Network of Intelligent Sound Agents. The compositions take advantage of the characteristics of the system, which is designed to monitor, predict and react to the performer's level of engagement. In the set of visual and written instructions comprising the scores, the network is fed with pre-composed musical motifs, stimulating responses from the system that relate to the inputted musical material. We also discuss some of the key challenges in applying traditional composition techniques to an intelligent interactive system.

1. INTRODUCTION

The relationship between music composition and new technologies has been always challenging to illustrate. Taking as an example the traditional instruments, they offer rich possibilities for music compositions. However, there has been a considerable difficulty in arriving at the same possibilities with digital systems that contribute to the development of new musical instruments. As the data processing abilities of technological devices advances at an exponential rate, most of the research related to new musical instruments has focused on technical aspects rather than aesthetic ones [1, 2]. Nevertheless, some grounds of interesting ideas for the process of composing for new systems have been discussed [3, 4]. In addressing the problematic area in this line of research, we recognise building new idiomatic writing for new musical instruments as crucial to the development of instruments that constitute new musical expressions.

Our contribution to this discussion is creating a set of musical compositions, exploring the use of motivic through-composition techniques in the context of a system with semi autonomous intelligent behaviour. Composing by applying variations of small pre-conceived musical ideas was championed by composers in the common practice period, which were then became an unquestionable element of the classical tradition [5]. As we seek for underlying alternative principles to provide us with a better understanding of composing for new musical instruments, we aim to build stronger bridges between classical music heritage and new musical interfaces. We apply this idea not only in the composition process itself, but as an integral part of the interaction and communication process between the elements of an interactive music performance system.

In this paper we present our compositions *"NOISA Études #1 and #2"* written for our previously presented interactive music performance system; *the Network of Intelligent Sonic Agents*, NOISA [6, 7]. NOISA proposed a solution for enhancing the average quality of the performers engagement (see Figure 1). We also highlight some of the key challenges in applying traditional composition techniques to an intelligent system and discuss alternative strategies to ratify the NOISA system as a valid instrument to perform electro-acoustic music composition in a live context.

Figure 1. General view of the NOISA system and instruments

2. RELATED WORK

The impact of idiomatic composition on a new musical instrument has been discussed in details by Tanaka [8]. The key concept relies on the fact that practitioners could construct a standard vocabulary of performance practice for novel instruments through repertoire, permeating the new instrument with an identity of its own. At this point, we also took inspiration from Boulez's ideas on correlations between innovation in music technology and aesthetic influences from the past to develop our composition strategies further [2].

Apart from applying motivic through-composition to inputted musical gestures, our reflections on Boulez's ideas led us to put in practice the concept of appropriation as coined in the visual arts history, *"artists tipping their hats to their art historical forebears"* [9]. Our application of appropriation in the sonic world is closer to what has been cataloged as *"borrowing"* [10]. It is a widespread practice in classical music rather than the more politically-oriented plunderphonics, [11] which is an statement against copyright. Therefore, we chose transforming sound sources that displayed a masterful application of through-composition techniques; pieces by Ludwig V. Beethoven, Eugène Ysaÿe, Modest Mussorgsky and Johann Sebastian Bach. In terms of the notation of the pieces, we used an effective visual system of position versus time, featured before in pieces such as Mikrophonie I, by Karlheinz Stockhausen.

Finally, we acknowledge the need of a close collaboration between designers and musicians for developing new instruments [2], both when considering the essential function of the musical material and in the peremptory need for building a repertoire. In that sense, we recognise the composition *"No More Together"*, which is based on social interactions of musicians through an interactive-spatial performance system [12]. We drew upon previous work in each area of intelligent interactive systems and aesthetic theories of appropriation to develop the compositions for the NOISA system, giving a first step into idiomatic musical writing.

3. THE NOISA SYSTEM

NOISA is an interactive system that consisted of three instruments for music creation, a camera with motion tracking capabilities, an electromyography armband and a central computer. The instruments' architecture is identical between them, differing only in the sonic result produced. Each instrument is built inside a black and white modular plastic box. The system is operated by manipulating ergonomic handlers, which can also position themselves through DC motors -when producing automatic musical responses-. As each handler is attached to a motorised fader, they provide position indicator and active position controller. While the right handler is in charge of sound production, the left one modulates the existing signal. In addition, we implemented the usage of the Myo armband and EMG data to provide dynamic control within with three levels of volume. An in-detail overview on how the system operates can be found in previous papers. [6, 7]

4. COMPOSITION PROCESS FOR A NETWORK OF INTELLIGENT AGENTS

As the new contribution of this paper, we composed two *études* upon the premise of feeding the NOISA system with brief musical *motifs*, in compliance with the motivic through-composition technique. We aimed to get an automatic response from the system as series of motivic variations in key moments that are dependant on the performer's predicted engagement level. The structure of the pieces also supports this goal, foreseeing an amount of automatic responses from the system.

Both compositions take advantage of the possibilities of the system in terms of attacks, dynamic and timbre. When interacting with the system, continuous movements allow continuous sound production, in opposition to short actions with silence in between. We used the Myo armband mainly for *crescendos* and *diminuendos*, emphasising climatic regions in the dramatic curve, which are also notated. Finally, when the handlers reach the top position, the instrument activates a *sustain* feature that generates textural *pedals* and complementary textures in the frequency spectrum when more than one NOISA instrument is operated in such a way.

4.1 NOISA Étude #1

The first piece, *"NOISA Étude #1"* is based on two *motifs* of contrasting characters of attacks and sustain. The gestures are measured in position versus time, with a total duration of 4 seconds each. The first one, *Motif I* is constituted by an specific rhythmic and melodic structure. Even though the position of the right slider is relative, it maintains the pitch relationship between the slider's movement in time as precise as possible. This is critical for the melodic identity of the *motif*. *Motif II*, on the other hand, portrays an identical downward motion in both sliders with textural characteristics rather than stating pitch content.

The performance instructions in the score ask the musician to get familiar with both *motifs*, exploring potential variations for each of them. This is necessary due to the structure of the piece in which there is freedom to develop own *motif* variations within estimated times and orders.

The first section of the piece encompasses the first 4 minutes, featuring a sequential feeding process to each agent with variations of the pre-established musical ideas. After this section is finished, the system will have analysed, stored and retrieved enough relevant information to become an active supporting interlocutor in its own right. In the second section, the performer is asked to wait for an specific amount of time after finishing a movement. The performer then follows a course of action depending if the system responds or not. If there is a response, the performer must provide a counter-response to the gesture retrieved by the system. The composition finishes by operating Agent 1, featuring short fragments of *Motif I* and always responding shortly to the gestures retrieved by the system. The ending of the piece is marked by a silence of automatic responses longer than 10 seconds. The first page of the score can be found in Figure 2 and a video of *NOISA Étude #1*, is available at https://vimeo.com/131071604

4.1.1 Audio Synthesis Module

Linking musical tradition with new technologies played a crucial role when selecting the sound sources to be manipulated. Our particular application of the *appropriation* concept consisted in radical transformations of fragments from music created with the motivic through-composition technique in mind. The chosen samples for the first piece were Modest Mussourgky's Pictures of an Exhibition in its transcription for Solo Guitar, Johann Sebastian Bach's Partita in A minor BWV 1013 for Solo Flute, and Ludwing Van Beethoven's Piano Sonata No. 21 in C major, Op. 53, also known as "Waldstein". All the sound processing happens in Pure Data.

The agents produce sound through sample-based granular synthesis, frequency-tuned. As each of the individual instruments is constituted by two sliders, we designed a multilayer interaction for each of them guaranteeing the production of complex sonic textures from a simple gesture input. Inside the granular synthesizer designer, the left slider, aka the sound producer, modifies the playback speed of every individual grain. At the same time, it controls

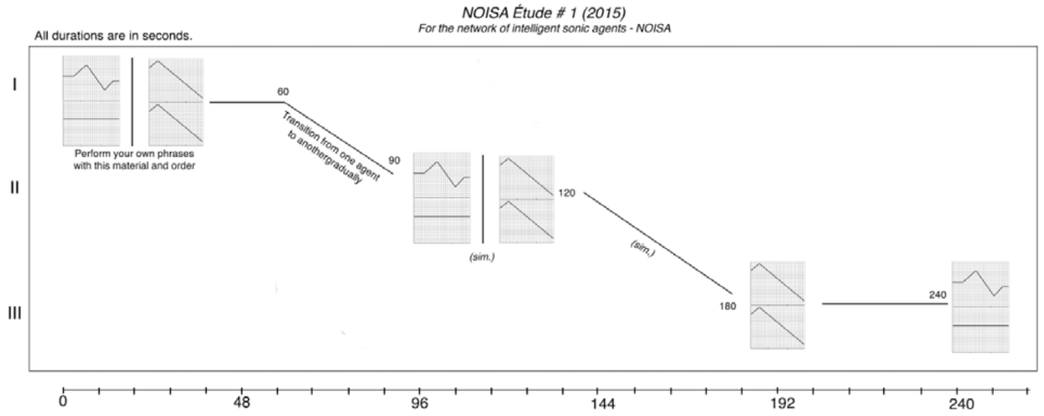


Figure 2. First page of the score for the *"NOISA Étude #1"*. The player is asked to compose previously phrases with the material given, and then follow a timeline of events. During the transitions, the instruments involved can be played simultaneously.

proportionally the wet/dry level of a fast *fourier* transform reverberation effect applied to a duplicate of the signal.

4.2 NOISA Étude #2

On the same vein, *"NOISA Étude #2"*, is a second set of performance instructions created to showcase the compelling, evolving and complex soundscapes possible with NOISA. Again, the system is fed with variations of a fixed musical motif, encouraging the system to recognise elements of the musical phrases and create its own set of versions emulating a human musical compositional process. Additionally, the Myo Armband is used in a creative way as an independent element for dynamic control, using raw data extracted from the performer's muscle tension.

In the score, there are three staffs per each of the boxes (see Figure 3), indicating both sliders' position in time, which is measured in seconds. After a series of gestures are inputted on the manner of the exposition of the motif, the performer is asked to wait for an automatic response. Next to this, there are different course routes depending on the systems response. The main difference with the first *étude* relies on a much stricter structure: Rather than providing freedom to develop *motifs*, each action is fully notated and linked to an specific moment of the piece, having more resemblance with a composition for traditional instruments.

A closer look to first section, until the 1:30 minute mark, demonstrates a series of phrases constructed out of performing variations of a small motif. An example provided shows the prime *motif* and how it was modified to build the phrase. (see Figure 4) A video of *NOISA Étude #2*, is available at `https://vimeo.com/134134739`

4.2.1 Audio Synthesis Module

The time stretching process with phase vocoders happens as well in Pure Data, and functions in the same way for all agents. Each time the handler is modified, the entire sample is heard with different time stretching values. This handler additionally controls the *reverb time* parameter of reverberation effect in an inversely proportional manner - The faster the speed, the smaller *reverb time* will be obtained, and viceversa.

The right slider, as with the first *étude*, controls the tape head rotation frequency emulator of a pitch shifter device. This process is applied only to the right channel, resulting with a stereo signal carrying both the original and transported signal. Finally, in terms of samples sources, both the Beethoven's sonata and Mussourgky's piece prevail, although the sonic result is dramatically different as expected due to the nature of the new audio processing module. On the contrary, the Bach's piece was substituted by Ysaÿe's Sonata No. 3 Op. 27 for Solo Violin.

5. CONCLUSIONS

In this paper we presented a novel approach for expanding the repertoire of new musical interfaces. We described in detail the application of our approach through the creation of two musical compositions. We also discussed the need for making further efforts in research regarding the aesthetic dimensions of new musical interfaces. We addressed the composition process in two ways: By virtue of utilising sound sources from the common practice period, and second, by crafting the pieces following the rules of the motivic through-composition technique.

We presented an in-detail analysis of a formative study in a different paper, with the aim of identifying the difference between NOISA and a random system of responses. The information extracted made us effectively anticipate in this context that the freedom given to the performer in *"NOISA Étude #1"* resulted in an increased number of responses from the system when compared to *"NOISA Étude #2"*. As in the latter the performer has to follow strict instructions, it seemed to require complete focus, making it not necessary for NOISA system to react towards improving the level of engagement. In opposition, giving certain creative freedom to the performer in *"NOISA Étude #1"*, had resulting spans with decreased engagement levels. Therefore, the NOISA system acted more vividly according to its designed behaviour.

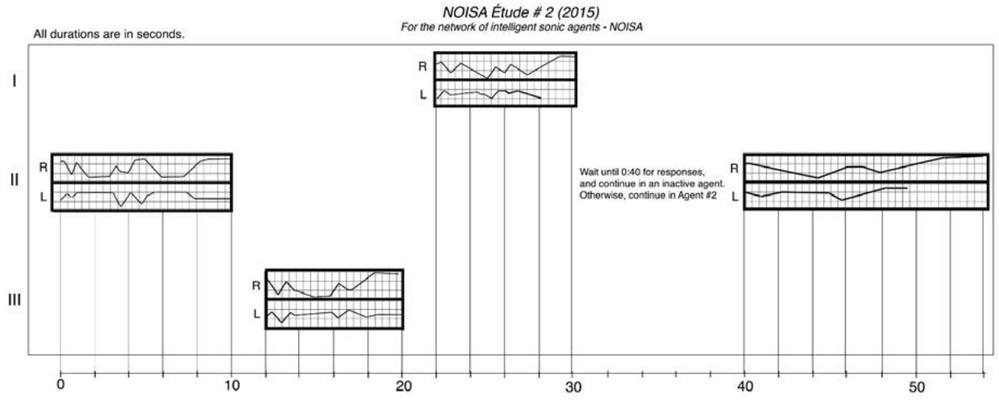


Figure 3. First page of the score of NOISA Étude #2. Durations and actions are notated in detail, while spaces without activity are silences.

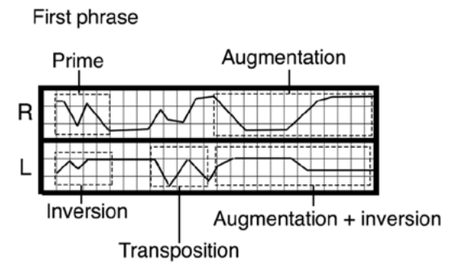


Figure 4. Graph showing some of the motivic variations contained in the first phrase of the piece.

Finally, we can note that these two compositions ratify NOISA as a valid instrument to perform electroacoustic music in a live context. We are planning to keep expanding the musical repertoire of the instrument by creating new work, as we develop the interactive system further on. We also hope to encourage other performers to experiment and perform with our system. The whole code comprising NOISA, including Pure Data patches and documentation, is available to download at `https://github.com/SopiMlab/NOISA`.

6. REFERENCES

[1] L. Landy, *Understanding the Art of Sound Organization*. MIT Press, 2007. [Online]. Available: https://books.google.fi/books?id=exwJAQAAMAAJ

[2] P. Boulez, "Technology and the Composer," in *The Language of Electroacoustic Music*. Springer, 1986, pp. 5–14.

[3] T. Magnusson, "Designing constraints: Composing and performing with digital musical systems," *Computer Music Journal*, vol. 34, no. 4, pp. 62–73, 2010.

[4] T. Murray-Browne, D. Mainstone, N. Bryan-Kinns, and M. D. Plumbley, "The medium is the message: Composing instruments and performing mappings," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011, pp. 56–59.

[5] W. E. Caplin, *Classical form: A theory of formal functions for the instrumental music of Haydn, Mozart, and Beethoven*. Oxford University Press, 2000.

[6] K. Tahiroğlu, T. Svedström, and V. Wikström, "Musical Engagement that is Predicated on Intentional Activity of the Performer with NOISA Instruments," in *Proc. of the Int. Conf. on New interfaces for Musical Expression*, Baton Rouge, USA., 2015, pp. 132–135. [Online]. Available: https://nime2015.lsu.edu/proceedings/121/0121-paper.pdf

[7] K. Tahiroğlu, T. Svedström, and V. Wikström, "NOISA: A Novel Intelligent System Facilitating Smart Interaction," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '15. New York, NY, USA: ACM, 2015, pp. 279–282. [Online]. Available: http://doi.acm.org/10.1145/2702613.2725446

[8] A. Tanaka, "Musical performance practice on sensor-based instruments," *Trends in Gestural Control of Music*, vol. 13, no. 389-405, p. 284, 2000.

[9] R. Atkins, *ArtSpeak: A guide to contemporary ideas, movements, and buzzwords, 1945 to the present*. Abbeville Press Publishers, 1997.

[10] K. Y. Peter, *Intellectual property and information wealth: issues and practices in the digital age*. Greenwood Publishing Group, 2007, vol. 2.

[11] J. Oswald, "Plunderphonics, or audio piracy as a compositional prerogative," in *Wired Society Electro-Acoustic Conference*, 1985.

[12] A. Parkinson and K. Tahiroğlu, "Composing Social Interactions for an Interactive-Spatial Performance System," 2013.

Exploiting Mimetic Theory for Instrument Design

Philip Wigham
Contemporary Arts, MMU
p.wigham@mmu.ac.uk

Carola Boehm
Contemporary Arts, MMU
C.Boehm@mmu.ac.uk

ABSTRACT

This paper will present a first instrument and discuss its design method, derived from principles informed by mimetic theories. The purpose of these design principles is to create new and innovative digital music instruments.

Even though mimetic theories are known to be important in the communication, engagement and expression of music performance, this ongoing enquiry represents the first consolidated effort to develop design principles from mimetic theories. [1], [2]

As part of the project, a development cycle is being followed to produce, evaluate and improve the design principles, and as part of this paper, a first prototype will be presented.

This paper covers a short description of the first prototype, describes the design process towards developing some generically applicable design principles and covers some of the underlying theories around empathy, communicative musicality and mimetic participation.

1. INTRODUCTION

This paper presents first outcomes and an initial prototype instrument, produced as part of a project that aims to develop instrument design principles informed by theories of communication and perception collectively referred to (in this paper), as mimetic theories. These theories include inter-modal perception [3], empathy [1], [2], [4], communicative musicality [1] and mimetic participation [2].

Existing digital music instrument (DMI) design theories have also been taken into consideration, looking at gesture [5], instrument efficiency [6], inevitability [7], affordances[8], [9] and Human Computer Interaction (HCI) [10], [11].

The first prototype was designed by applying these mimetic theories to the existing DMI theories, guiding the choice of features, instrument shape, materials and mapping of controls to synth parameters. We began with the premise that if design principles were to be developed that took mimetic theories into consideration the production of instruments following these principles should ideally improve what Trevarthen & Malloch have coined as communicative musicality [1] of the instrument (see chapter below). Thus an effective mimetic instrument should be successfully employed/exploited in therapeutic, community music and/or performance/audience contexts.

2. PROTOTYPE DEVELOPMENT

Figure 1. Prototype 1.

The first prototype (Figure 1, the first of three so far) was developed to explore the initial premise of these principles. All prototypes have some basic features that can be found in many gesture-based instruments and that allow simultaneous control of independent parameters. Basic features include a range of sensors to accommodate the independent manipulation of several controls simultaneously, as well as controlling the initiation, length and pitch of the notes. The integration of physical body/movement gestures rather than limiting gestures by using knobs, buttons and faders, allows a full range of small, medium and large gestures creating a much wider range of gestural movement to control the sounds.

The first design (Figure 1) was based around the guitar. A version of Delalande's classification of gesture [12], modified by applying mimetic principles, has been used to develop the gestural elements of the prototype:

2.1 Initial Gesture

Initial gestures begin the sound wave transient, and are quite often percussive. With an acoustic instrument's sound this transient is often important for the recognition of the timbre [13]. Daniel Levitin gives this description, "The gesture our body makes in order to create sound from an instrument has an important influence on the sound the instrument makes. But most of that dies away after the first few seconds. Nearly all of the gestures we make to produce a sound are impulsive" [14].

When a gesture is seen to be initiating the sound the two senses of sight and hearing are working together to create a perception of the instrument being played. Depending on the movement of the gesture it could be possible to either enhance the audience's perception of the instrument or conversely reduce its impact, by intentionally subverting the natural expectation of the audience. For example, the visual expectation of the audience, when hearing louder sounds, might be to see larger movements, which in acoustic instruments would be the case, but in electronic instruments could be inverted. In this situation smaller movements creating louder sounds might confuse an audience.

If further such subversion to audience expectancies is created the sounds being heard may no longer be perceived as being connected with the instrument on stage. If the listeners do not connect the sounds with the instrument then they may not be able to imagine creating those sounds on the instrument themselves. Therefore the affect of mimesis would be greatly reduced.

2.2 Modulating Gesture

Modulating gestures are gestural movements that occur after the sound has been initiated, modulating parameters that affect the sound in some way. Synthesisers generally have many parameters that may be changed during the sound production, and so there are several modulating gestures to complement these synth parameters. These modulating gestures may be split into three sizes: small, medium and large. Small gestures are difficult to see but affect the sound; medium gestures can be seen from a small distance; large gestures are movements that can be seen from distance.

As with the initial gesture, a compliance or subversion of expectation, using common parameters, such as pitch-bend, could have similar effects as discussed above. However, parameters that affect the sound in new ways, not analogous to an acoustic counterpart, may not be treated in the same way by the listener. The new sound and connected gesture may intrigue the listener with its uniqueness and unfamiliarity. This may allow new associations to be made with the instrument and how it should be played. This could lead to interesting relationships between the gesture and synthetic sound, and provide informative movement that enhances the sound rather than remaining abstract and detached from the aural information.

In conjunction with the initial gestures, carefully designed modulating gestures should strengthen the mimetic impact.

2.3 Inter-Modal Gesture

Inter-modal gestures include all components/features that do not affect the sound but have a visual presence. Although the gestures do not directly change the sound, taking the McGurk effect [3] into account, they influence the perception of them.

An important inter-modal consideration is in the way that the instrument looks and feels. The first prototype was created to look more like an acoustic instrument than a typical controller. It is made mainly from wood and great effort has been made to hide the technology where possible. This is not only so the performer may feel more like they are performing with an acoustic instrument, but also so that listeners may be given the impression of an acoustic musical instrument similar to a guitar.

Creating an 'acoustic' look to the instrument should elicit a mimetic response in the audience, allowing them to form an impression of the mechanics of the instrument.

3. PROTOTYPE 1

With consideration to mimetic theory the aforementioned guitar based design gives the observer a starting point from which to understand the performance motions and gestures. This allows an initial understanding of the controller and a basis to build in new gestures specific to the device.

The initial gesture requires the 'plucking' to initiate the sound, and the positioning of the 'fret board' hand to alter the pitch. This will be familiar enough for guitarists to immediately pick up the controller and begin playing with an intuitive sense of control, but will also be familiar enough for non-guitarists to gain a modicum of control with little effort.

The prototype utilises a variety of sensors to exploit the various movements that are possible with a guitar-based instrument, producing modulating gestures that control synthesiser parameters. Sensors placed at fret and bridge positions can detect small modulating gestures, mapped to appropriate synthesiser parameters.

Other sensors detect medium modulations from the hands, and large modulations from movements of the controller. Guitarists will be familiar with these larger gestures but in most cases, on an electric guitar they will be inter-modal, (not actually affecting the sound). On the controller they are modulating gestures, and are mapped to additional synth parameters.

The concept of mimesis is an interesting one to consider when analysing the performer-audience relationship. However, this concept allows us to furthermore align instrument design not only to the creative aims of performers or instrument makers but to address specifically parameters that might be considerably involved in allowing audiences to feel that performed music on digital instruments is accessible to them. It is apparent through this research, so far, that the inclusion of mimetic theories during the design and development of controllers will open up interesting avenues for new devices.

There is a compromise between a shape suited to synthesised sounds and one influenced by theories of mimesis that will promote mimetic participation. The guitar

base should afford users to know how to initially generate sounds, which should in turn improve mimetic understanding of the instrument thereby enabling mimetic processes.

The design is also an attempt to balance many facets of instrument design: a unique digital instrument/controller vs. traditional acoustic form; small nuance based performer orientated gestures vs. large spectacle audience orientated gestures; ease of play for beginners vs. complexity of play for mastery; simplicity of design and use vs. complexity and flexibility of control.

These design facets are pulled together with the common thread of mimetic theory, including empathy, communicative musicality and mimetic participation.

4. MIMETIC THEORIES

There are many relevant areas of research important to instrument design, such as affordance, gesture, inevitability and efficiency [6]–[10]. However the main thrust of research for this project has come from three key areas: empathy, communicative musicality, and mimetic participation.

4.1 Empathy

Empathy is intrinsic to the mimetic process. Trevarthen and Malloch [1] describe how musical mimesis may facilitate improved social empathy. Communicative musicality produces an empathy and understanding between mother and baby [15]. This imitative process is essential to creating empathy. How we understand each other and the way we communicate involves empathetic, mimetic response. Cox states that 'part of how we comprehend music is by way of a kind of physical empathy that involves imagining making the sounds we are listening to' [2].

Empathy and sympathy are key processes in communicative musicality, which Malloch [15] describes as 'movement that allows mother and infant to express themselves in ways that are sympathetic with the other'.

4.2 Communicative Musicality

Trevarthen's [16] studies of the earliest interactions between newborn babies and their mothers, known as motherese or proto-conversation have been shown by Malloch [15] to contain patterns, repetitions, rhythms, pitch and intonation variations which are very musical in nature. Trevarthen's collaboration with Malloch suggests that the presence of this 'communicative musicality' between mother and baby is essential for healthy social and cognitive development of the child [1], [15], [16].

This innate, imitative ability is utilised throughout our lives to communicate, empathise and to make sense of the world around us. We understand music and performance through this visceral 'empathy', wanting to 'join' in through mimetic participation.

4.3 Mimetic Participation

Mimetic participation can be used to describe how we understand and imitate a process such as playing a musical instrument. It can be an uninvited urge to copy someone or join in such as tapping your foot or humming to music [17]. Arnie Cox asks 'Do you ever find yourself tapping your toe to music?' and then suggests that 'Informally conducting, playing 'air guitar', and 'beat boxing' (vocal imitation of the rhythm section in rap) are similar responses' [17].

Through researching the mechanisms of empathy and communicative musicality it should be possible to emphasise/exaggerate the effect of mimetic participation, creating instruments that invoke their 'air' cousins in audience/listeners.

5. DESIGN PROCESS AND FUTURE EVALUATION

Figure 2 below shows the development cycle to be followed to produce, evaluate and improve the design principles.

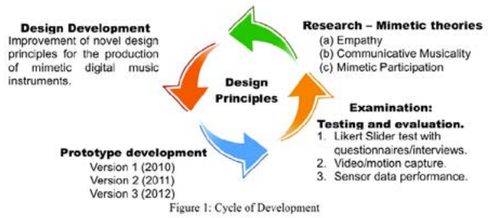


Figure 2. Cycle of Development.

The design process of the cycle of development includes taking measurements at live performance events as well as using video interviews. Analysis of video footage and audience/performer sensor data provides additional data sets, to compare mimetic designed instruments with traditional instruments.

An instrument with greater mimetic effect should elicit more imitative gestures. To test this hypothesis design principles are developed throughout the research project using multiple iterations of the above process. All prototypes are created using these principles and will be examined as described below.

Cox suggests that 'For many if not most of us, and for most kinds of music, music nearly demands mimetic participation (overt or covert)' [2]. Cox's 'covert imitation' involves imagining physical actions and 'overt imitation' refers to outward movements or gestures such as tapping your feet [2]. This overt/covert mimetic participation will be examined during a series of performances.

A composition using modulated, synthesised sounds will be carefully composed so that it can be performed identically, using the same sound generator, by both standard keyboard controller and prototype. The composition will be performed to a click track ensuring consistency in performance and time stamping for data analysis. Separate performances of this composition, one using keyboard, another the prototype, will allow a comparison of the 'mimetic' features of the prototype and their associative non-mimetic gestures of the keyboard.

At each performance the performer and audience (approximately 20 people) will be videoed to allow comparisons of specific performance gestures and audience response. This video footage will show medium/large mimetic gestures of the audience, such as 'air' guitar type motions. Time stamped data from movement, force and vibration sensors arranged around audience seats will be analysed for small/medium gestures such as foot/finger tapping.

Due to the nature of covert mimesis, it will be necessary to interview audience members to understand their thought processes during the performance. Video interviews will be undertaken after each performance to discover how each audience member felt they were affected by the performance and if they had any desire to imitate or join in. The interview videos will also be analysed to look for imitative gestures used in the interviews.

Interviews will be retrospective and reliant on the interviewee's memory. However an additional Likert[1] Slider test will be implemented during the performances. Before the performance, each audience member will be provided with a physical slider. They will be asked to move the slider during the performance from 1 to 10, in response to an appropriately designed question, such as how much they would like to join in with the performance, and/or how engaged they feel with the performance. These sliders will produce data that is time-stamped so the values can be compared with the other data/video analysis.
Once the data/video has been analysed it can then be used to compare the differences/similarities between the keyboard and prototype performances, looking to see if the features of the prototype have an increased mimetic effect causing greater imitation and desire to join in.

6. CONCLUSION

We believe that even though mimetic theories are known to be important to the communication, engagement and expression of music performance, this ongoing enquiry represents the first consolidated effort to develop design principles from mimetic theory. Our initial prototypes point towards the validity of the assumption that an instrument designed with mimesis in mind should elicit more imitative gestures.

This project, which is in the middle of the first iteration, will demonstrate a development cycle that produces, evaluates and improves the design principles, which are the core output of the PhD project.

These mimetic design principles will be tested and developed initially using progressive versions of the first guitar-based prototype design, following the development design cycle (Figure 2). A future paper will cover the results of these tests and the following iterations of design. To further develop the design principles, new and different mimetic prototype designs will be created and tested.

Acknowledgements
The authors would like to thank the Department of Contemporary Arts at Manchester Metropolitan University for supporting this research and development project.

[1] A Likert scale is a psychometric scale used in questionnaires, named after its inventor Rensis Likert.

7. REFERENCES

[1] S. Malloch and C. Trevarthen, "Musicality : Communicating the vitality and interests of life," in *Communicative musicality: Exploring the basis of human companionship.*, S. Malloch and C. Trevarthen, Eds. New York: Oxford Univesity Press, 2009, pp. 1–11.
[2] A. Cox, "Embodying Music : Principles of the Mimetic Hypothesis," *Soc. Music Theory*, vol. 17, no. 2, pp. 1–24, 2011.
[3] H. McGurk and J. Macdonald, "Hearing lips and seeing voices," *Nature*, vol. 264, no. 5588, pp. 746–748, Dec. 1976.
[4] T.-C. Rabinowitch, I. Cross, and P. Burnard, "Long-term musical group interaction has a positive influence on empathy in children," *Psychol. Music*, vol. 41, no. 4, pp. 484–498, Apr. 2012.
[5] J. W. Davidson, "She's the one: Multiple Functions of Body Movement in a Stage Performance by Robbie Williams," in *Music and Gesture*, A. Gritten and E. King, Eds. ASHGATE, 2006.
[6] S. J. Puig, "Digital Lutherie Crafting musical computers for new musics ' performance and improvisation," 2005.
[7] T. Machover, "Instruments, interactivity, and inevitability," in *Proceedings of the 2002 conference on New Instruments for Musical Expression*, 2002.
[8] R. L. Cano, "What kind of affordances are musical affordances ? A semiotic approach," in *L'ascolto musicale: condotte, pratiche, grammatiche. Terzo Simposio Internazionale sulle Scienze del Linguaggio Musicale.*, 2006.
[9] A. Tanaka, "Mapping Out Instruments , Affordances , and Mobiles," in *NIME 10*, 2010, pp. 88–93.
[10] M. Billinghurst, "Gesture Based Interaction," *Haptic Input*, pp. 1–35, 2011.
[11] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human Computer Interaction*, 3rd ed. Essex: Pearson Education Limited, 2004.
[12] M. M. Wanderley and B. W. Vines, "Origins and Functions of Clarinettists' Ancillary Gestures," in *Music and Gesture*, A. Gritten and E. King, Eds. Aldershot: Ashgate Publishing Limited, 2006, pp. 165–191.
[13] S. Malloch, "Timbre and Technology: An Analytical Partnership," *Contemp. Music Rev.*, vol. 19, no. Part 2, pp. 155–172, 2000.
[14] D. J. Levitin, *This is Your Brain on Music: Understanding a Human Obsession*. Atlantic Books, 2008.
[15] S. N. Malloch, "Mothers and infants and communicative musicality," *Music. Sci.*, vol. 3, no. 1, pp. 29–57, 1999.
[16] C. Trevarthen, "Learning about Ourselves, from Children: Why A Growing Human Brain Needs Interesting Companions?," 2004.
[17] A. Cox, "Hearing, Feeling, Grasping Gestures," in *Music and Gesture*, A. Gritten and E. King, Eds. Aldershot: Ashgate Publishing Limited, 2006, pp. 45–60.

Viewing the Wrong Side of the Screen in Experimental Electronica Performances

Sonya Hofer

ABSTRACT

While there is considerable attention in music and media studies on works that jump to the screen, from MTV, to Blu-ray ballets, to the Black Swan, to videogames, in this paper I will look instead at works that jump behind the screen, the laptop screen. In most experimental electronica performances, the laptop computer is the main "live" instrument. In this mode of performativity, not only are our performers situated behind a screen, a figurative "curtain"—or literally the backside of the screen—becomes what is viewed in the live setting, offering a curious perspective on mediatized musical contexts.

The most pervasive critique of experimental electronica performances stems from a perceived lack of visual spectacle and gesture by the performer, whose main "live" instrument is the laptop computer. Performances are sometimes read as lifeless, disengaged, tedious, effortless, and automated. In this mode of performativity, not only is the artist situated behind a screen, so too is the audience. In a live setting, audiences view the backside of the screen, offering a curious perspective on mediatized musical contexts.

The laptop is central to the conception and experience of experimental electronica, with direct and clearly articulated qualitative consequences.[1] For this reason, and the very fact that I write this paper on a laptop, my project is to delve deeper into our meaningful relationships with laptops by thinking more holistically and phenomenologically about screens. I consider "screenness" within the context of musical performance, here examining notable live sets by acclaimed experimental electronica artist Tim Hecker. Typical of experimental electronica, Hecker's performances take place in a range of contexts and, in what follows, I look at two very different sets. Closely evaluating each and taking cues from their critical reception, I employ screenness as a mode that frames our experiencing of the music, impacts our assumptions and expectations about laptop performativity, and also reveals how the music effectively works in dialogue with and within its varied, live musico-experiential contexts.

Montreal-based Tim Hecker is one of those musician-artist creators who is at once a DJ, electronic musician, and sound artist. He emphasizes how much more dynamic, diverse, and challenging this field of creative activity has become by pointing out the possible limitations and entanglements of such labels. As his work straddles these not-so-discrete fields, it raises issues of liveness and mediation, notably the distinction between how music is presented live and how it is *present* live.[2] Additionally, his tools play a highly affective role in how people conceive of "live" music. Many experimental electronica musicians, like Hecker, use the laptop as their main instrument in both creation and performance. Ideas about authenticity are embroiled in ideas about technology, and are in continual flux, as popular music and media scholar Andrew Goodwin has observed:

> Playing analogue synthesizers is now a mark of authenticity, where it was once a sign of alienation—to pop iconography, the image of musicians standing immobile behind synths signified coldness … Now it is the image of a technician hunched over a computer terminal that is problematic—but that, like the image of the synth player, can and will change…[3]

The use of the laptop as a "problematic," as Goodwin notes, has to do with how it is viewed in this relational-historical context. For example, oftentimes laptop DJs are compared unfavorably to DJs who use turntables, though the turntable itself was once regarded with skepticism. As an instrument, turntablists and turntablism eventually came to be validated. Yet, where a DJ is seen hopping back and forth among the dynamic turntables with hands and body in motion, the laptop performer, in a false analogy, is more often viewed as immobile and cold, concentrating on a stationary box while one hand makes miniscule mouse clicks. This sense that experimental electronica performers aren't "doing" much is imbricated with deep-rooted ideals regarding what constitutes musical skill. The use of software such as Max/MSP and Ableton, which allow for increased automation and simulation, contributes to a misconception that laptops make musical creation and performance "easier," or even that the music has been faked. And while Goodwin acknowledges the legitimacy of programming skills, his evocative choice of words insinuate the pervasive regard of computer-wielding laptop artists as technicians, i.e., not even as musicians.[4]

The already-numerous critiques of laptop performativity, while not my direct focus, offer me a spring-board for considering a different critical lens on laptop-based experimental electronica performances. These critiques, including Ed Montano's aptly titled article "'How Do You Know He's Not Playing Pac-Man While He's Supposed to Be DJing?': Technology, Formats and the Digital Future of DJ Culture" give me license to reiterate the question and to answer that indeed we don't know if the DJ is playing Pac-Man.[5] What interests me is how Montano's question encapsulates the differing ways we value "liveness" and what we demand from live performance. The very fact that it matters whether or not the laptop DJ is playing Pac-Man indicates how audiences require visual confirmation of the artistic process. Audiences want to know that something, and furthermore *someone*, is creating sound in real-time.[6] Of particular interest is how these critiques put the screen at the center. The screen consumes performers to an apparent detriment. The screen, not even its front, but its back, is an audiences focal point. Screens make the source and nature of sound production (or Pac-Man playing) opaque, and direct our gaze onto something visually mundane.

Artists have responded, consciously and subconsciously, in varying ways to "remedy" the issues accompanying laptop performance, from simply moving about more, to adding a visual component like an accompanying film or immersive lighting, to incorporating additional live musicians or more traditionally viewed instruments, to projecting screenshots to the audience, among others.[7] Even when these responses were not necessarily an explicit attempt to amend critiques of laptop performativity, they might nonetheless be viewed as remedies helping audiences to "engage" with the music, for example by offering a greater sense of bodily connection to the sound and sound production; providing something to look at; and, in general, making a "musical process" clear. This paper, however, does not suggest what types of laptop-centric shows are "better" live, or seek to prove whether or not DJs are really playing Pac-Man. It does not ratify or repudiate the so-called "problems" of laptop performativity, or prescribe how audiences "should" recalibrate and adapt to these performances. Instead, taking a different critical tack than others' qualitative critiques of laptop performativity, I focus on how the laptop screen signals and plays a significant role in constructing contextual meaning.[8] With the aim to enrich these critical discourses that address such issues of performativity and reception, I examine them within "screenness" as an expressive and experiential paradigm in analyzing the performances.

The laptop is not merely an inert object. As part of a generation of screen natives, for whom the growing presence of the laptop and mobile devices are key, I consider how we are conditioned to them, suggesting a kind of fundamentality to them, a screenic technogenesis.[9] I build from theories of screen subjectivity, notably Kate Mondloch who writes about screenness in experiential art contexts, saying:

> The viewer-screen connection is a site of radical inter-implication: it includes the projection screen and other material conditions of

---

[1] I emphasize here the actuality of public performances and as they are mediated through YouTube. People can experience experimental electronica in other contexts, for examples, at home, in their cars, by themselves with an iPod, listening to LPs in a café, which would lead into interesting continued projects, i.e., considering screenness in these contexts.

[2] I have written on this topic in greater depth, considering the work of Richard Chartier, see: "'Atomic' Music: Navigating Experimental Electronica and Sound Art Through Microsound," *Organised Sound* 19/03 (Dec 2014): 295-303, as a kind of companion piece.

[3] Goodwin cited in: Philip Auslander, *Liveness: Performance in a Mediatized Culture*. London and New York: Routledge, 1999, 11. In addition to Goodwin's words here, by foregrounding "cultural and historical contingencies," I point to Auslander's key writings on the condition of mediation and its impact on "liveness," which help me to introduce and to begin framing issues in approaching Hecker's music and laptop performativity more generally.

[4] This bifurcation between technician and musician as a kind of critique in electronic music has many historical roots. See, for example, Georgina Born's observation of this in her ethnography of IRCAM. See: Georgina Born, *Rationalizing Culture: IRCAM, Boulez and the Institutionalization of the Musical Avant-Garde*. Berkeley: University of California Press, 1995.

[5] Ed Montano, "'How Do You Know He's Not Playing Pac-Man While He's Supposed to Be DJing?': Technology, Formats and the Digital Future of DJ Culture," *Popular Music* 29, no. 3 (2010): 397–416.

[6] Montano and Auslander have both addressed this topic, i.e., how a sense of authenticity created is rock music (Auslander) and electronic music (Montano) by making a musical process visual on stage.

[7] See for examples, Fennesz who has collaborated closely with various video artists as part of his live set, Mouse on Mars who has added a drummer on drum kit as part of their live set, Squarepusher who at times performs on his electric bass, and also Slub who have incorporated screen projections of their live coding as part of performances.

[8] Others who have written on issues of laptop performativity, with differing aims include: Montano, discussed above; Nick Collins, who voices concern that many complexities in programming by a performer may be "lost" on audiences and seeks new modes of communicating such aspects in, "Generative Music and Laptop Performance," *Contemporary Music Review* 22, no. 4 (2003): 67–79; Timothy Jaeger, who offers a pointed critique of laptop performers whom he characterizes as not provoking new paradigms for performance, see "The (Anti-) Laptop Aesthetic," *Contemporary Music Review* 22, no. 4 (2003): 53–57; and Tad Turner, who cites a lack of comportmental code due to a diversity in venue, and seeks a mediation of, and adjustment to, the various strengths and weaknesses of venue types in "The Resonance of the Cubicle: Laptop Performance in Post-digital Musics," *Contemporary Music Review* 22, no. 4 (2003): 81–92.

[9] Here I allude to Hayles' foundational writings on the fundamentality of human/technological intersections. See, notably: N. Katherine Hayles, *How we think: Digital media and contemporary technogenesis*. Chicago; London: The University of Chicago Press, 2012.

> screening, but also encompasses sentient bodies and psychic desires, institutional codes, and discursive constructs.[10]

As such, I want to look at Hecker's performances as part of a screened worldview, thinking about how the computer screen specifically activates a particular kind of intimate psychological and physiological relation. Its materiality and presence are key, not simply what it screens, inclusive of sounds, but in the very mode of screening itself, conjured in places and presupposed as a frame of reference. Lucas Introna and Fernando Ilharco, who offer a phenomenology of screens, foreground this approach, writing:

> …we do not want to focus on the experience of watching screens, nor do we want to focus on the content of screens. We want to suggest that there is something prior to all of these, namely that which conditions us to turn to it 'as a screen' in the first instance.[11]

In considering screenness, the following looks at two recent performances by Hecker, uploaded to YouTube by audience members, thus positioning ourselves also as part of an audience. Immediately upon viewing these performances, one is alerted to three distinct intertwined phenomenological relationships and perspectives, which I will touch on throughout and focus on towards the end of this paper: Hecker's relation to the screen; the audience's relation to Hecker's screen; and our relation to the musical scene via our own screens through YouTube.

The first video depicts Hecker's performance at the 2012 Pitchfork Music Festival in Chicago. We see here his performance of the track "Virginal I," which was at the time of this performance still unreleased, but later appeared on his seventh full-length album, the 2013, *Virgins*. This track segues into "The Piano Drop," from his 2011 album, *Ravedeath, 1972*. In some ways this might be an expected scene. In the context of a festival set, Hecker is focalized, framed through a stage setup, which is further amplified by the meta-screen through which we watch on YouTube. However, what are the peculiarities of this mode of performativity as evidenced here?

Example 1: YouTube video of Tim Hecker at 2012 Pitchfork Music Festival: https://www.youtube.com/watch?v=8gkpp7dn2j8[12]

As Introna and Ilharco note, screens call us to attention, as what is framed in the screen is already contextualized as important. Media theorist Lev Manovich sharpens this point, emphasizing how the screen is also antagonistic: in displaying what is included, infers what is excluded, i.e., to screen is also to choose.[13] Yet a corresponding friction is not so much about the audience's longing or necessity to see the obstructed content, but about the nature of the obstruction itself. The laptop screen here is not a frame, window, or mirror; rather, drawing from the etymology of "screen," the laptop is a barrier that obscures Hecker to the viewers, and simultaneously obscures the audience from Hecker.[14] As the following YouTube commenter laments of the video:

> I have trouble with stuff like this... I LOVE his record, but couldn't imagine it being an enjoyable experience to see him mess around behind a laptop...[15]

This kind of exclusion is alienating, like watching a date text from their mobile phone at the dinner table—a comparison that perhaps gives more depth to critics' concerns regarding the lack of engagement of audiences and performers in laptop performativity. Yet, the backside of the screen, something that we encounter perhaps just as frequently as the front-face, is drastically under-theorized.

I would argue that these backsides are, crucially, part of the potentiality of screens, otherwise we would not decorate mobile phone cases, or place tape over the Apple logos on the backs of our laptops as Hecker does here. Even if its backside is supposed to be inconsequential, it does not disappear, rather it remains as a significant visual cue of the screen. Screens, as Mondloch and Anne Friedberg have theorized, construct "an architecture of spectatorship."[16] Laptop performances are fascinating precisely because they undermine the "coercive nature of screenviewing;" we are not drawn into the laptop screens' flashing images, content, and fantasy, but it still crucially frames and directs our viewing.[17] A screen signals that we should be watching, but faced with the screen's backside, and compounded further by the frame of the YouTube video, audiences pose the question: What should we be watching?

In staged musical performances, we are not typically suspicious about what we should pay attention to while absorbing the music. There may be many things to watch, or maybe we choose to close our eyes and watch something else in our minds, but we are not generally pushed to question: *wait, what should I be watching?* But the screen, and particularly its backside, leads audiences to consciously or unconsciously pose this as a significant critical question, as they search for what to attend to. A focal point is pre-supposed by the screen, but the focal point is confounded. Intensifying this, the musical sounds are dislocated from the screen, which further perplexes our relation to the screen and scene as a focal point. And finally, an untethered sense of the performance, prompted by conditions surrounding screenness, is unintentionally bolstered by our presumed subject, Hecker himself, as I will discuss shortly.

In further examining this performance, one assumed condition of laptop performativity is an engagement between the performer and the laptop. Thinking about this condition fits neatly into ideas about screen subjectivity more broadly, and also, subsequent complaints concerning laptop performances. The presumed orientation of a viewer to a screen as they are habitually drawn in, from mobile devices to films, is generally "face-to-face," which elicits Goodwin's characterization of the hunched computer performer who is engrossed with the screen rather than engaged with the audience.[18] We come to expect someone using a laptop, and thus a laptop performer, to be occupied with the screen in this manner. Yet, is Hecker absorbed by the laptop in the way we assume performers will be, or as screenness presupposes?

Re-watching the Pitchfork set, in this video, Hecker takes a more flexible stance and is at times only peripherally engaged with his laptop. While that is partially explained by the fact that he has other instruments to attend to, there is still a kind of peculiarity to this scene. Can we imagine a pianist getting up and walking away from their instrument mid-performance? Or a vocalist talking to techs mid song? Although certainly in rock/pop sets there may be some dialogue between the musicians and stage technicians, this is usually quite discreet.[19] Should Hecker be wholly engrossed in his terminal? Again, a laptop performer is commonly assumed to be consumed by their own screen, and this is a common complaint about their performances. Hecker goes against our expectation of how a performer interacts with a screen. However, while he is not as singularly locked into his screen, this does not counteract that complaint of laptop performativity, rather his performance seems even further destabilized as a performance. Here his engagement with his screen suggests something more quotidian rather than performative—perhaps akin to the way I can take a break from typing this paper to put away my coffee cup all the while still "working." In a similar sense, this performance, and the video of it, can be mystifying for both the live audience and for YouTube viewers. Is he playing music or is this part of the soundcheck? Again, what should I be watching here? Should I be seeing this? How should I be seeing this? What should I be seeing while listening to this? Would I know this was part of the "actual" set if I happened upon this scene?

These issues and questions regarding Hecker's set as a performance are compounded in the YouTube video itself. With Hecker further framed through the screen and already contextualized as a focal entity, as suggested by Introna and Ilharco, the sense of what we are and should be watching is doubly unclear.[20] The YouTube viewer comments communicate the precariousness of this performance and allude to well-worn complaints about laptop performativity:

> I feel he needs a more interesting set, with jaw dropping lighting and visuals, something to help intensify the experience.[21]

> Yeah. I agree. This seems awkward. This isn't daytime music.[22]

> This seems like the absolute worst place for an artist like Tim Hecker to play. I feel bad for him really. He should be playing a dim lit theater, a cave, a church, an alleyway.... anything but a day time outdoor festival set.[23]

Examining screenness with regard to Hecker's Pitchfork set has led us to question what we are viewing as a performance by amplifying a disjuncture between what one experiences aurally and what one experiences visually. Similarly, the YouTube comments draw attention to this disjuncture, but their critiques do not simply react to a problem of the visuals—a lack of visuals, the wrong visuals, or uninteresting visuals. Rather, I suggest these critiques reveal how certain

---

[10] Kate Mondloch, *Screens: Viewing Media Installation Art*. Minneapolis: University of Minnesota Press, 2010, 4.

[11] Lucas D. Introna and Fernando M. Ilharco, "On the Meaning of Screens: Towards a Phenomenological Account of Screenness," *Human Studies* Vol. 29, No. 1 (Jan., 2006), 58.

[12] Video upload from "snivelttam." *Tim Hecker – Virginal I/The Piano Drop – 2012 Pitchfork Music Festival,* https://www.youtube.com/watch?v=8gkpp7dn2j8, 2012.

[13] Introna and Ilharco, 68 and Lev Manovich, *The Language of New Media*. Cambridge: MIT Press, 2001, 94.

[14] I allude to longstanding discourses considering "the frame" from Leon Battista Alberti's "window" to Edward T. Cone's "musical frame." More specifically I draw on Anne Friedberg's noteworthy text considering, situating, and (re)appraising the screen among metaphors of the window. Anne Friedberg *The Virtual Window: From Alberti to Microsoft*. Cambridge: MIT Press, 2006.

[15] From "David Needham," https://www.youtube.com/watch?v=8gkpp7dn2j8, 2014.

[16] Mondloch, 23, emphasizes this idea, extending ideas from Friedberg.

[17] Mondloch, 24 and others, see for example: Jeffrey Sconce, *Haunted Media: Electric Presence from Telegraphy to Television*. *Durham*: Duke University Press, 2000.

[18] While complicated by certain mobile device practices, "face-to-face" frontal orientation, as the expected orientation has been examined in depth, See: Ingrid Richardson, "Faces, Interfaces, Screens: Relational Ontologies of Framing, Attention and Distraction," *Transformations* 18 (2010).

[19] I would note that there are, at times, pressing matters in which the stage technicians would need to take immediate and direct action on the stage. From what we know contextually about Hecker's set, there was a rainstorm approaching, and perhaps they were preparing the equipment. However, this does not detract from how we (in the audience at Pitchfork and through the YouTube video) might understand how Hecker is able to interact with his screen.

[20] Introna and Ilharco, 66-67.

[21] From "CanadianCombatWombat," 2014.

[22] From "Christopher Robin," 2013.

[23] From, "Jestin Jund," 2012.

conditions of screenness in this festival context direct observers to parse unnaturally between the sonic and the visual, rather than appreciate the set as more holistically "experiential." Moreover, and significantly here, the critiques also point to how laptop performativity and screenness are deeply conditioned by and enmeshed in place and context. In the following, I take up the last comment as a call to question and consider one of Hecker's performances that does take place in an actual church.

While I have ruminated on Hecker's performance as a parsing of the sonic and visual, as is also implicit in the viewer commentary, my aim here is not to separate the senses. As we will see in the following performance, thinking about screenness may have different affective ramifications in this context, moving beyond the visual vs. sonic, toward a more holistic sense of experientiality. This video documents a 2012 performance that took place in the Chiesa di Santa Cristina in Parma, Italy, a beautiful baroque church that is unassuming from the outside, yet resplendent inside. Hecker here performs tracks mainly from his 2011 *Ravedeath, 1972* album, itself primarily recorded in the Fríkirjan Church in Reykjavík, Iceland, where he used its pipe organ as central source material.

Example 2: YouTube video of Tim Hecker in the Santa Cristina Church, 2012: https://www.youtube.com/watch?v=_UcW4aSLQBQ[24]

Viewing this video of Hecker's concert, I ask, what is striking about it? Immediately, what strikes me is that someone would record and then upload a 49 minute visually static video of a live performance—a performance in which Hecker, the headliner, is completely omitted.

In watching this video, I would then ask, what might it say about the position of the performer in the live setting? Perhaps the response is blunt: since there is "not much" to watch, arguably there is no need to focus the camera on Hecker. One might then wonder why a video would be taken at all if the scene is static i.e., why not just record the performance as sound? Or why not include some better-quality image (for example, there are many audio uploads on YouTube of sound recordings with subsequently added higher resolution visuals)? Another, perhaps more reflective response, considers the possibility that the video communicates the experience of how the music interacts with the space. Hecker does not appear in the video, but it is by no means devoid of rich and affective imagery—imagery that evokes and emphasizes the importance of the specific place and time of the performance. In this way the video communicates a centrality of experience and sense of place conjured quite differently in this context.

As Mondloch illuminates in her analyses of screen-reliant art installations, a screen calls attention to the "'real' space of the projective situation," that is, its actual surroundings, rather than being simply illusory (i.e., what is projected).[25] Within the church, one is not uncomfortably standing in a bruised field, waiting for imminent rain amongst a clutter of simultaneous musical performances, seeing a haphazard stage attended to by technicians, and ultimately presented with the "wrong" side of the main instrument engaged nonchalantly by the musician. In the church performance, rather than prompting viewers to scan for what to watch or to question what one sees, the screen potentially incites a different impulse and response. The YouTube comments accompanying the Chiesa di Santa Cristina video serve as evidence: there are no complaints regarding the concert, the performance, or the experience.

In this context, the screen makes people conscious of the place of the event in a way that is very different from the Pitchfork Festival. Much like a television set situated on the wooden floor of a white-walled gallery projecting a work of video art, Hecker's laptop screen amplifies a sense of place. While the screen may prompt the work, the screen, performance, Hecker, and *the work* are a part of a greater totality in which the audience is also entangled. The screen's materiality matters: its form, its design, its look, its front, its back, et cetera. Our imagination of a screen matters, our mental and physical relation to it matters, its contextualization matters.[26] In experimental electronica performances, such as Hecker's, this potentiality of the laptop as a totality works in dialogue with the music. Synchronously, his musical works are in dialogue with place.[27] Hecker's music both conjures and creates places that might be experienced through the act of listening. Hecker's music dimensionalizes sound, constructing and idealizing ambiances, and works in consolidation with the laptop, which is in part a vessel for this sound, but crucially also an important signifier for how we experience the music in place.[28]

Evoking the platial, the experiential, I am led to consider the multi-dimensionality of sensory experience, and to extend the many critical debates in screen and media studies that question hierarchies between the sonic and the visual. Indeed in such experimental electronica performances, as with Hecker's in the Chiesa di Santa Cristina, neither the sonic nor the visual may be a priori.[29] This is not to deemphasize how one sense can be foregrounded or stimulated, but rather, to draw attention to how the senses merge, moving toward what Kay Dickinson has theorized as synaesthesia.[30] Hecker's performances seem to break out of a simply audiovisual frame. The sense of immersion afforded by a heightened multi-sensory experience allows the audience to participate *with* and *as* the work.[31] Making this point more obvious is Hecker's 2010 performance at the Big Ears festival, in Knoxville's Tennessee Theater, where he performed in total darkness. This is not to say that visual blackness was, or is ever, blank, or to imply a continued conflict between vision and sound by eliminating former, but to suggest how smell, touch, vibration, and imagination, are ever-present and elemental to musical experience. Similarly, Michel Chion, prolific theorist on the sensoriality of film, observes that experiential contexts that deemphasize the visual allow for an interesting kind of paradox:

> …having only sound at my disposal has stimulated me to create rich sensations that are no longer just sound sensations but also tactile and trans-sensory. I have often observed that when there is nothing but sound, the sound becomes all the sensations and ceases to be "just" sound.[32]

A screen works reciprocally with places informing our perception of experience, and while all performances are multisensory by nature (as all existence is), there is something different between the Pitchfork and Chiesa performances. In the case and context of the former, the laptop seems to splinter and bar, while in the latter it invites coalescence and aesthetic sensoriality. Different places and contexts spurn different aesthetic experiences in which screenness plays a dynamic role.

To take a final turn, I consider how we are now, this moment, outside of the actual performative places. While it alters the meaning of how the screen and Hecker's music emphasize the platial, it is critical to focus more acutely on how these performances are mediated and remediated as YouTube videos. Moreover, this distinction in observing the performance through YouTube may also serve as a kind of extension to Philip Auslander's seminal writings, i.e., how mediatization indeed draws attention to a sense of liveness. These videos certainly shape my impressions in reception; they offer the screen via the screen. As I mentioned previously, the Pitchfork video redoubled the effect of the stage, seeming to reinforce and reiterate how Hecker and his laptop were assumed to be central to a frame of presentation. And while we do not view Hecker and the laptop in the Chiesa di Santa Cristina video, the video's screenness and corresponding construction of place seems doubly manifest in Hecker's absence. The video's perspective in the Chiesa set implies the presence of the performer by so consciously directing away from him, simultaneously highlighting how viewers participate in the piece and meaning-making. The audience member-videographer directs away from Hecker and the laptop, imparting meaning into the video. Perhaps, as I infer as a fellow spectator, the videographer means to echo and to invoke the phenomenological sense of place for us; perhaps the videographer's perspective implies how they view Hecker and his corresponding mode of laptop performativity, i.e., not assumedly as central, necessary, or interesting to the frame, and thus directs elsewhere. The screenic perspective opens up a fascinating critical power in spectatorship.[33] Furthermore, in contrast to the numerous biting comments regarding his Pitchfork set, viewers who commented on the Chiesa video praise the live performance, saying: "This is brilliant, thanks for

[24] Video upload from "brokenbywhispers." *Tim Hecker live Santa Cristina Church, Parma, 02-11-2012,* https://www.youtube.com/watch?v=_UcW4aSLQBQ, 2012.

[25] Mondloch writes extensively on this topic in her book, citing the writings of art critics Cornwell, Krauss, and Michelson and artworks by Michael Snow and VALIE EXPORT. Mondloch, 61.

[26] I echo Mondloch's writings clearly here on this topic. Mondloch's writings on the materiality of screens are key here, as she emphasizes how "interface matters" when considering screen-reliant art installations and extrapolations of meaning. Her writings along with Introna and Ilharco, discussed throughout, are central in considering screens and "screenness" in this manner.

[27] His works feature performance places and exploit the nature of specific rooms and halls. Accordingly, many are treated and exhibited as sound installations. There is an audible physicality to his music: allusions to sound as dynamic "masses" and their cultivation over time; having sound "objects" appear and disappear; differences of and play with contrasting sonic textures such as thick/thin, foreground/background; evocations of color; sonic effects that recall objects, their surfaces, their boundaries; all in essence, quite visceral qualities.

[28] My inquiry into screenness and musical performance is in many ways in line with one of audiovisual studies' most significant questions concerning how media contexts dialogue with platial surroundings. See Richardson and Gorbman "Introduction," *The Oxford Handbook of New Audiovisual Aesthetics*. Eds. Richardson, Gorbman, Vernallis. Oxford: Oxford University Press, 2013, page 25.

[29] The divergent standpoints of E. Ann Kaplan and Andrew Goodwin, with the latter critiquing the former as diminishing the sonic, are suggested here. See: Kaplan. *Rocking Around the Clock: Music Television, Post Modernism and Consumer Culture*. New York: Routledge Press, 1987; and Goodwin. *Dancing in the Distraction Factory*. Minneapolis: University of Minnesota Press, 1992.

[30] See Kay Dickinson "Music Video and Synaesthetic Possibility," in *Medium Cool: Music Videos from Soundies to Cellphones*. Durham: Duke University Press, 2007.

[31] See Richardson and Gorbman, pg. 7, who have keenly observed a mode of sensorialty in cinema and aptly cite Michel Chion, who wrote foundational texts on the sensual audiovisuality in film (See: *Audio-Vision: Sound on Screen*, and *Film, a Sound Art*). Alluding an aforementioned "frame" above, I consider Monica E. McTighe's relevant book, *Framed Spaces: Photography and Memory in Contemporary Installation Art*. Hanover: Dartmouth College Press, 2012.

[32] Michel Chion, "Sensory Aspects of Contemporary Cinema," in *The Oxford Handbook of New Audiovisual Aesthetics*. Eds. Richardson, Gorbman, Vernallis. Oxford: Oxford University Press, 2013, 325.

[33] Citing Anne Friedberg, Mondloch underscores the critical power of media viewing, Mondloch, 56-58.

sharing—must have been an amazing experience."[34] They do not express any sense of "loss" regarding Hecker's absence from the video, nor does it fall short of what constitutes satisfactory performativity. Perhaps the YouTube viewer experiences the performance, not simply as mediated through the video, but also as remediated, that is, fundamentally already existing through a lens of screenness.[35] Enacting screenness, I would argue that the construct of the video as a static shot then indeed makes sense, as it already implies a screenic perspective whereby an orientation toward Hecker's screen as a presumed focality was confounded and then averted. But, this diverting does not speak detrimentally to the performance, rather quite conversely, it speaks directly to it being a platial, multi-sensory, holistic experience.

Sparked by the deeply contested viewpoints regarding performativity in experimental electronica, I offer a different framework for engaging with the music. This paper offers one way of viewing these performances, using the screen as a central hub for extrapolating meaning, as the presence of the screen has an effect on how people experience music in place and vice versa. It might be argued, that I place too much focus on this one element. I would point out, however, that re-focusing on many of the other experiential parameters—for example, on gestures, timbre, or lighting would still draw us back into discussing the role of the laptop. This approach continues in the direction of theorists who have written extensively on the inescapably mediated nature of performance, taking into consideration the laptop, its screen, and screenness as one rich avenue for examining laptop performances among their very diverse musico-experiential contexts. In doing so, I hope to contribute more nuanced and dynamic understandings in modes of performativity and spectatorship with regard to new media, and specifically here, of experimental electronica performances, which have been so widely and unevenly critiqued.

REFERENCES

[1] Philip Auslander. *Liveness: Performance in a Mediatized Culture*. London and New York: Routledge, 1999.

[2] Jay David Bolten and Richard Grusin. *Remediation: Understanding New Media*. Cambridge: MIT Press, 1998.

[3] Georgina Born. *Rationalizing Culture: IRCAM, Boulez and the Institutionalization of the Musical Avant-Garde*. Berkeley: University of California Press, 1995.

[4] Video upload from "brokenbywhispers." *Tim Hecker live Santa Cristina Church, Parma, 02-11-2012,* https://www.youtube.com/watch?v=_UcW4aSLQBQ, 2012.

[5] Comment from "CanadianCombatWombat." https://www.youtube.com/watch?v=8gkpp7dn2j8, 2014.

[6] Michel Chion. *Audio-Vision: Sound on Screen*, Ed. and Trans. by Claudia Gorbman. New York: Columbia University Press, 1994.

[7]----------*Film, a Sound Art*. Trans. by Claudia Gorbman and C. Jon Delogu. New York: Columbia University Press, 2009.

[8] Nick Collins. "Generative Music and Laptop Performance," *Contemporary Music Review* 22, no. 4 (2003): 67–79.

[9] Kay Dickinson. "Music Video and Synaesthetic Possibility," in *Medium Cool: Music Videos from Soundies to Cellphones*. Durham: Duke University Press, 2007.

[10] Comment from "faultelectronica." https://www.youtube.com/watch?v=_UcW4aSLQBQ, 2013

[11] Anne Friedberg. *The Virtual Window: From Alberti to Microsoft*. Cambridge: MIT Press, 2006.

[12] Andrew Goodwin. *Dancing in the Distraction Factory*. Minneapolis: University of Minnesota Press, 1992.

[13] N. Katherine Hayles. *How we think: Digital media and contemporary technogenesis*. Chicago; London: The University of Chicago Press, 2012.

[14] Sonya Hofer. "Atomic Music: Navigating Experimental Electronica and Sound Art through Microsound," in *Organised Sound* Vol. 19/3 (December 2014): 295-303.

[15] Lucas D. Introna and Fernando M. Ilharco, "On the Meaning of Screens: Towards a Phenomenological Account of Screenness," *Human Studies* Vol. 29, No. 1 (Jan., 2006).

[16] Timothy Jaeger. "The (Anti-) Laptop Aesthetic," *Contemporary Music Review* 22, no. 4 (2003): 53–57.

[17] Comment from, "Jestin Jund." https://www.youtube.com/watch?v=8gkpp7dn2j8, 2012.

[34] From "faultelectronica," https://www.youtube.com/watch?v=_UcW4aSLQBQ, 2013.

[35] I cite ideas of remediation, the evocation and representation of one medium in another, as theorized by Bolten and Grusin in their key text *Remediation: Understanding New Media*. Cambridge: MIT Press, 1998.

[18] E. Ann Kaplan. *Rocking Around the Clock: Music Television, Post Modernism and Consumer Culture*. New York: Routledge Press, 1987.

[19] Lev Manovich. *The Language of New Media*. Cambridge: MIT Press, 2001.

[20] Kate Mondloch. *Screens: Viewing Media Art Installation*. Minneapolis: University of Minnesota Press, 2010.

[21] Monica E. McTighe. *Framed Spaces: Photography and Memory in Contemporary Installation Art*. Hanover: Dartmouth College Press, 2012.

[22] Ed Montano. "'How Do You Know He's Not Playing Pac-Man While He's Supposed to Be DJing?': Technology, Formats and the Digital Future of DJ Culture," *Popular Music* 29, no. 3 (2010): 397–416.

[23] Comment from "David Needham." https://www.youtube.com/watch?v=8gkpp7dn2j8, 2014.

[24] Ingrid Richardson. "Faces, Interfaces, Screens: Relational Ontologies of Framing, Attention and Distraction," *Transformations* 18 (2010).

[25] Michel Chion, "Sensory Aspects of Contemporary Cinema," in *The Oxford Handbook of New Audiovisual Aesthetics*. Eds. Richardson, Gorbman, Vernallis. Oxford: Oxford University Press, 2013: 325-330.

[26] Comment from "Christopher Robin." https://www.youtube.com/watch?v=8gkpp7dn2j8, 2013.

[27] Jeffrey Sconce. *Haunted Media: Electric Presence from Telegraphy to Television*. *Durham*: Duke University Press, 2000

[28] Video upload from "snivelttam." *Tim Hecker – Virginal I/The Piano Drop – 2012 Pitchfork Music Festival,* https://www.youtube.com/watch?v=8gkpp7dn2j8, 2012.

[29] Tad Turner. "The Resonance of the Cubicle: Laptop Performance in Post-digital Musics," *Contemporary Music Review* 22, no. 4 (2003): 81–92.

Balancing Defiance and Cooperation: The Design and Human Critique of a Virtual Free Improviser

Ritwik Banerji
Center for New Music and Audio Technologies (CNMAT)
Department of Music
University of California, Berkeley
ritwikb@berkeley.edu

ABSTRACT

This paper presents the design of a virtual free improviser, known as "Maxine", built generate creative output in interaction with human musicians by exploiting a pitch detection algorithm's idiosyncratic interpretation of a relatively noisy and pitchless sonic environment. After an overview of the system's design and behavior, a summary of improvisers' critiques of the system are presented, focusing on the issue of balancing between system output which supports and opposes the playing of human musical interactants. System evaluation of this kind is not only useful for further system development, but as an investigation of the implicit ethics of listening and interacting between players in freely improvised musical performance.

1. INTRODUCTION

"…a so-called 'pitch follower' — a device known to exercise its own creative options from time to time…"
— George E. Lewis, [1]

Since George Lewis' *Voyager* [1, 2], researchers in computer music have designed a variety of virtual performers of free improvisation [3, 4, 5, 6, 7, 8, 9, 10, 11, 12], interactive music systems built to perform (ideally) as just another semi-autonomous musician in an ensemble of human improvisers. As Lewis writes regarding *Voyager* [2], virtual improvisers should listen and respond to human playing to produce sonic output which appropriately shifts between supporting and opposing other improvisers' musical ideas. At one extreme, the improviser should feel the influence of their playing on the system's output, or as Michael Young describes it, a sense of "intimacy" in the human-machine interaction [12]. At the other extreme, the player should neither feel that the system simply mirrors their behavior, nor the need to "prod the computer during performance," as Lewis puts it. Similarly, Young describes this as the "opacity", or the lack of transparency in the relationship of system input and output.

However, building a system to interact with human improvisers with a sense of intimacy and sympathy is a special challenge given the tendency of free improvisers to explore timbrally complex material. As several system designers have noted, free improvisational practice frequently features unpitched and noisy sounds and generally avoids musical structures based in pitch, such as melody or harmony. Likewise, in order to design systems which listen and respond sympathetically to such playing, many researchers have discarded any design approach in which the system's real-time analysis of the human player's sound output is solely pitch-based [5, 6, 7, 8, 9, 10, 12]. In addition to pitch, such systems use a variety of spectral analysis tools to decompose the complexity of common practice in free improvisation into several components, such as noisiness and roughness [10].

Given how improvisers often play, this is a logical direction to follow in building systems to exhibit greater "intimacy" in human-machine interaction. Nevertheless, it overlooks the hidden utility of the pitch detector's interpretation that pitchless sounds, such as a styrofoam ball scraped against a snare drum head, the hum of an amplifier, or a woodwind multiphonic, have a definite pitch, or more often, are simply a run of pitches. While the pitch detector's interpretations of these pitchless sounds as "pitched" is technically inaccurate, they may provide a means realizing Young's ideal of opacity in the system's input-output transformation, a sense of mystery and individualism which Lewis wittily depicts as the algorithm exercising "its own creative options".

This paper presents the design of a virtual free improviser, known as "Maxine" [4], built to creatively exploit the pitch detector's odd interpretations of sonic material in free improvisation, an auditory environment for which it is destined to fail. The system capitalizes on the pitch detector's simultaneously "intimate" and "opaque" interpretations pitchless sounds in order to produce an overall interactivity which balances between sympathetic and oppositional behaviors. After an overview of system design and its resultant behavior, the paper concludes to focus on the critical evaluation of improvisers who have played with the system as part of an ethnographic study on social and musical interaction in free improvisation.

Ultimately, soliciting practitioners' feedback on such systems is not simply about improving design, but actually deepening understanding of musical practice on a social-scientific and musicological level. Asking players to critique the behavior of a non-human musician allows them to be far more explicit about what they really expect of other players in socio-musical interaction. Because improvisers wish to show respect for the musical liberty of their peers, it is a social taboo for them to explicitly express such preferences to each other [13]. Conversely, because this system is a mere machine, players feel no risk of infringing upon the musical liberty of others by offering critical feedback to the designer [14]. As a result, testing such systems with improvisers provides a unique opportunity for them to candidly express expectations they hold of their peers' behavior which they are hesitant to ever make explicit in their routine social interactions.

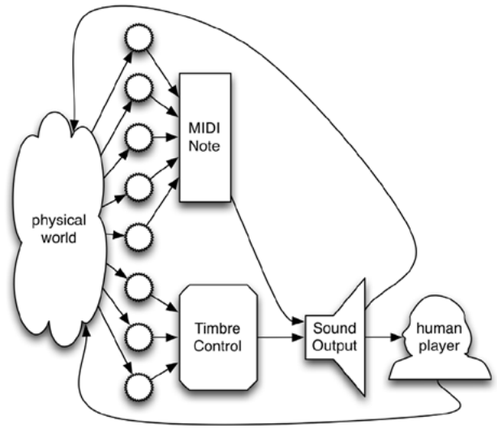


Figure 1. Overall flow of information through system, from physical world, through agents, to sound output and the human performer, and back to the physical world. Small gears represent each agent.

2. SYSTEM DESIGN

The system uses a multi-agent architecture. Several identical agents simultaneously process auditory input and control sonic output. Agents operate non-hierarchically and in parallel to the rest (see Figure 1). While each agent is the same, internal values and end outputs can vary significantly at any given time. Each agent functions as a single "arm" or "finger" of the system, controlling either MIDI note or controller values based on the processing of auditory input from its respective "ear." The rest of section 2 traces the flow of information through a single agent (see Figure 2).

2.1. Input and Feature Extraction

The system receives audio input from the physical world into two dynamic microphones (Behringer XM8500), one aimed at the human performer and the other at the system's own loudspeaker output. Analog to digital conversion occurs through a MOTU Ultralite audio interface and audio feature extraction occurs in Max/MSP. Each agent extracts three basic features from incoming digital audio signal in real-time: 1) pitch and 2) attack information, both from Tristan Jehan's [pitch~] object [16] (based on Puckette's [fiddle~] [17] and hereafter referred to simply as "[pitch~]"), and 3) amplitude information.

2.2. Note Event Timing Control

Agents use changes in pitch and attack information reported by [pitch~] to control output timing. Reported pitch changes and attacks are sent to a timer, which measures the interval between reported events, regardless of whether the event was a pitch change or an attack. These durations are used to set the agent's *base quantization*, (BQ). Similar to the *tatum* [18], or temporal atom, BQ is the shortest duration for any MIDI output from the agent. Actual durations, or local quantization (LQ), are a random multiple of the BQ between one and 15. However, not all attacks or reported changes in pitch are used to set the BQ, and reporting of these events is filtered by a probability gate (see section 2.4.).

Commands to change the LQ to a new random multiple of the BQ are sent out at the rate of the current BQ. Another probability gate, however, only allows a percentage of these commands to cause an actual change in the LQ. Similarly, note output messages are being sent out at the rate of the LQ, but another probability gate controls the percentage of these note output messages resulting in an actual MIDI message.

2.3. Pitch Selection

Note output from each agent is selected from a three-value pitch-set within a three octave range (C1 to C4) which may change at any time. A note P_{xc} in the pitch-set $P_{(1,\ 2,\ 3)c}$ is changed, to a new value (randomly chosen within same range as above), P_{xn} when an incoming value from [pitch~] P_I is a match (or when $P_I = P_{xi}$). Essentially, this mechanism is much like the modern mechanical arcade game "whack-a-mole". However, similar to the BQ, not all matches ($P_I = P_{xc}$) trigger a change in the agent's pitch set because of filtering out by another probability gate.

2.4. Probability Gates

Probability gates determine the likelihood that an incoming message will be passed through the gate. This probability rises and falls according to incoming amplitude from the microphones. However, the mapping is constantly changing both in direction (inverse vs. direct) and in degree (see Figure 3).

Specifically, current incoming volume is scaled to probability according to current high (H) and low thresholds (L) for scaling. Changes in threshold values are triggered by changes in pitch reported by [pitch~]. However, this triggering is also passed through the very same probability gate. Once a change in threshold is triggered, current incoming amplitude data is polled at the rate of the

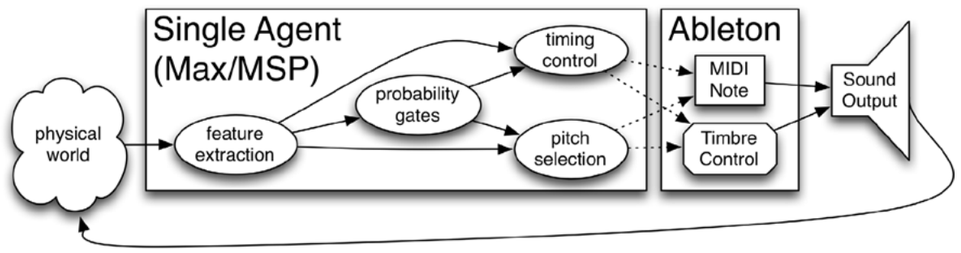


Figure 2. Flow of information through a single agent, from physical world to sound output (and back to the physical world.)

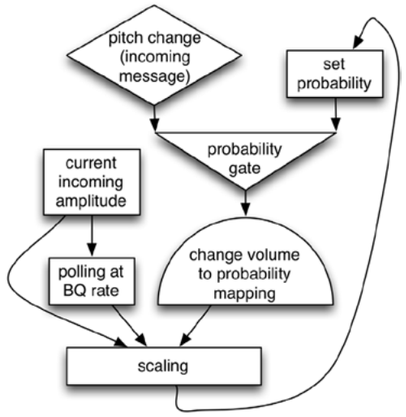


Figure 3. Internal structure of probability gates

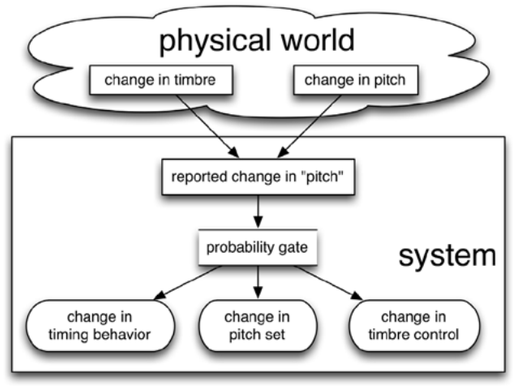


Figure 4. Effect of changes in timbre and pitch in the physical world on system behavior.

current base quantization and sent as the new high or low threshold. When the "high" threshold is lower than the "low" threshold, an inverse mapping of volume to probability results.

2.5. Sound Output and Timbre Control

Based on the above, each agent sends either MIDI-note or -controller values from Max/MSP for output in Ableton Live. In typical performance practice, five agents are responsible for note generation while three control the manipulation of timbre parameters in Ableton. Controller values are used to manipulate the timbre of virtual instruments in Ableton Live. Sound outputs from Ableton Live typically include metal percussion, synthesized versions of prepared or "extended" guitar and piano techniques, a variety of synthesizers, and signal processing tools (e.g. filters, delay, etc.) used to control audio feedback. Strictly speaking, agents controlling timbre only send MIDI notes to Ableton (see Figure 2). These note values are set to control timbre parameters using Ableton's MIDI-mapping capability.

3. SYSTEM BEHAVIOR

3.1. Intimacy and Opacity from [pitch~]

Because of its reliance on pitch detection as its primary means of real-time analysis of sonic input, the system's interactive behavior *simultaneously* exhibits both the opacity and intimacy Young idealizes.

3.1.1. Intimacy

[pitch~] offers as a robust, if crude, means of providing this virtual improviser with a real-time analytical representation of its sonic environment. As noted in section 2.2., pitch changes (as reported by [pitch~]) are used to control the relative temporal density of the system's note output. This allows the system to follow the overall event-density of the human performer (e.g. if the human player's event durations are between 100-400ms, the system's output will be in a similar range.)

3.1.2. Opacity from Intimacy

However, regardless of whether incoming sounds are pitched or not, [pitch~] guesses the "pitch" of these sounds, looking for an even spacing of harmonic partials in order to identify a fundamental frequency. Again, it is senseless to say that a noisy, aperiodic sound like, that of a styrofoam ball scraped against a drum head, has a definite *pitch.* Defiantly, [pitch~] defiantly claims otherwise, reporting an unsettlingly specific value: (hypothetically) a clear D#4, exactly 17.9 cents sharp!

This interpretation is *both* intimate and opaque. It is *intimate* in that for a given spectral profile, [pitch~] will always produce the same estimated pitch value. It is *opaque* in that the relationship of this value and the sound itself, given its pitchless quality, seems almost random.

The system uses [pitch~]'s simultaneously opaque and intimate interpretation of such sounds to produce behavior which is both sympathetic but also mysterious in its interactive logic. For example, given the noisiness of transients at the onset of many note events, [pitch~] often parses such sounds as first a flurry of rapid "pitch" changes (see Figure 4) and then the pitch audibly produced. The human player plays just one note, but the system may respond disproportionately, reacting with a gust of activity to this small perturbation. Still, because of the use of probability gates (section 2.4.), the system does not always react in this unbalanced manner.

For complex, time-varying timbres, [pitch~] allows the system to react temporally to the human player's subtle modulations of sound. As a player's timbre changes, [pitch~] yields a new estimated "pitch" value, effectively allowing the system to use [pitch~] as a crude approximation of spectral flux [19, 20], or the degree of variation of spectrum over time. In turn, this enables the system to vary its timbral output and control note activity (section 2.3.), in a manner corresponding to the overall pacing and event-density of the human player.

3.1. Feedback Effects

The combination of the whack-a-mole mechanism for pitch selection (section 2.3.) and feedback (section 2.1.) implies that the system may trigger itself to change its own pitch set. For example, if the system is producing Gb4 while also waiting for Gb4 to appear in the environment, one would assume that the system has a high likelihood of causing itself to change its pitch set. In practice this rarely occurs. Much of the system's output features heavy manipulations of timbre. As a result, while the system is sending a Gb4 note value to Ableton, the timbre of this Gb4 may be so significantly manipulated that [pitch~] detects no Gb4 from the input signal.

Overall, the use of feedback, in which at least microphone is directed at the system's own output, allows the system to demonstrate a better balance between resistive and cooperative interactivities. In my own early experiments with this system (as a saxophonist), I felt the need to "prod" the system, as Lewis would say, with microphones were only aimed at my instrument. To correct this, the current feedback setup was implemented. This allows the system to respond in a more satisfyingly unpredictable manner as it reacts to its own output, or even the slight hum of the loudspeaker itself. System sounds, alongside other environmental noises, are, like anything picked up by the microphones, just registered as more "pitches" and also stimulate the system to respond. In testing the system with improvisers, microphone placement was often varied based on the player's preference for a more aggressive or sympathetic interactivity.

4. EVALUATION OF THE SYSTEM BY FREE IMPROVISERS

4.1. Methodology

The system was first designed in 2009. Since then, over 90 musicians, primarily in Berlin, San Francisco, and Chicago, have played informal, private improvisation sessions with this system. The initial motivation for testing was to solicit the critique of players with extensive experience in free improvisation in order to identify directions for further development of the system. I approached improvisers directly, as I myself was also a free improviser on the saxophone.

Musicians were asked play a series of duets with the system and give their commentary on its behavior immediately after each piece. These duets were usually between five and ten minutes, though often longer. Pieces typically ended in the same way that most freely improvised pieces do: performers are silent for a period of time and then look up to indicate that the piece is over.

After each piece, I let the improviser lead the conversation, allowing them to focus on whatever they found most interesting or problematic about the system's behavior. As I have discussed elsewhere [14], taking an open-ended ethnographic approach to researcher-subject interaction and letting the subject drive the conversation is far more effective than using pre-determined questions and quantitative evaluation in a controlled laboratory setting. Given the wide range of behaviors desired and resultant from free improvisation (whether human or machine), specific questions and criteria may be irrelevant to the interaction which just transpired and hinder the performer's discussion of their subjective experience of it. As a result, the next section only focuses on performers' comments which directly referred to the issue of balancing support and opposition, and thus is a small cross-section of the over 300 hours of commentary on this system collected over several years.

This method not only elicited performers' critiques of the system, but their discussion of similar moments of frustration with human players. In other words, asking them to critique the system brought them to express not just what they expect of a *machine*, but what they expect of other *people.* This methodology makes performers feel safe to express such socio-musical expectations in a manner that they never experience in face-to-face interaction with other players. Again, respecting the musical liberty of their peers, improvisers tend to avoid negative critical discussion of their peers' playing. After all, if the practice of free improvisation purports to emancipate musicians from the "rigidity and formalism" [21] of other musical practices, it makes no sense for players verbally express their expectations to other performers, whether beforehand or afterwards.

By stark contrast, players found that critiquing a non-human musician enabled them to articulate expectations that they normally feel implicitly barred from openly expressing in their normal social interactions with other improvisers. While improvisers feel that such direct expressions of expectation are essentially a taboo practice in their socio-musical world, this hardly means that no player has specific expectations, much less that no other player disappoints them. As one performer put it, "I *wish* I could tell other people things like this!"

4.2. Summary of Results

4.2.1. Preference for Greater Assertiveness

On the one hand, many players found the system to be too meek, hesitant, or reserved in its interactive behavior. These performers felt that the system did not take enough initiative in interaction, or as one player put it, failed to "inspire" them. They found themselves stifled by the system's dependence on human input and its tendency, in their experience with it, to wait for the human player to play before producing material of its own.

For example, one player found that the system's silence in some situations was not experienced as a polite gesture of yielding to others, but as a frustrating inability to sustain the drama of the interaction. Such behavior reminded him of an inexperienced improviser whose reticence and lack of confidence saps an improvisation of its overall energy. In response to this sort of system behavior, he stressed the critical importance, in his view, of simply taking a risk and playing *something* rather than remaining silent because of indecision or self-doubt.

For another player, the problem was not that the system was too quiet, but rather that it was *too sensitive* to his playing. Rather than remaining with one sonority for a period of time, the system reacted to his playing too frequently, causing it to change its timbral output too rapidly for this player's tastes. He described the system's behavior as "fickle", even childlike, flitting from idea to idea. While he found moments of the system's behavior interesting, its inability to stay with one sonic idea was disappointing. Instead he would have preferred greater obstinacy in the system's improvisatory behavior, remaining with one idea and allowing the human player to go elsewhere sonically without reacting immediately to each

new idea he introduced. For him, this behavior would have allowed for a more meaningful contrast, juxtaposition of sonorities, or tension to develop.

4.2.2. Preference for Greater Sensitivity

Nevertheless, many other players found the system too aggressive. One individual directly blamed this on the fact that the system reacts to itself. This mechanism and its effects gave him the feeling that the system behaved like a self-absorbed individual during improvisation, following its own ideas rather than cooperating with others. Similarly, another player described this as a failure to "meet me halfway", or the inability to choose material which partially emulated, and partially deviated, from the choices of the other musical interactant.

In one rather illustrative case, the system persisted with a repetitive undulating feedback effect for nearly two minutes. During this time, the human player experimented with a variety of ideas (melodic runs, sustained tones, quick high-energy blasts, etc.). At one point he stopped playing and stared at the amplifier with a disgusted look, as if to tell the system, "stop!" Indeed, after the piece he described the system's behavior as "annoying" in its failure to sense his disgust for its playing at that moment.

4.3. Preference for Defiance? Two Individuals

Strikingly, two individuals showed a strong preference for defiant or resistant system behavior. For one Berlin-based cellist, playing with the system was a relatively comfortable experience. Though I asked him if we could do a quick initial piece just to check the volume balance between him and the system, he played with the system without pause for nearly an hour. Such a reaction is unusual, with most players preferring to play with the system for a much shorter period of time, in most cases no longer than twenty minutes.

In the follow-up conversation, he found that he liked playing with the system, but gave some curious reasons for his preference. What he enjoyed most about the interaction was the feeling that the system did not "really *listen*," as his put it. I was perplexed by this seemingly backhanded compliment. Later on, however, he explained his irritation with players, especially younger musicians, who tend to immediately respond to his playing with material that references (i.e. reproduction or mimicry) what he just played. By contrast, the system's inability to do so made him feel more comfortable.

The preference for defiant and resistive playing is all the more intriguing in the experience of another Berlin-based musician, this time a trumpet player. As discussed in section 3.4., when a player expresses a desire or interest in more or less aggressive playing, I experiment with varying microphone setups. In three pieces with this trumpet player, each of approximately ten minutes, three configurations were attempted: two microphones on the trumpet player, one on the trumpet player and one on the system, and two on the system.

Surprisingly, he preferred the configuration in which both microphones were aimed at the loudspeaker, away from the bell of his trumpet. Again, this preference, like that of the cellist, suggests a preference for a kind of musical interactivity which is more resistive than supportive. He preferred a manner of playing in which the other player would be listening closely at all times, but would not necessarily announce and demonstrate their awareness of the other player by immediately and unambiguously reacting to each new idea.

Like the cellist, he explained this preference as a capacity of interaction habit often avoided by younger players. He explained that when he was a younger improviser he too used to play in a much more reactive manner. However, as he became older and more experienced, he reduced this highly reactive tendency in his playing and, similarly, sought to work with players whose modes of interaction with other players were less obvious, or more "opaque," as Young might describe it [12].

4.4. Discussion

Unfortunately, commentary generated from extensive tests of this system with a variety of players ultimately gives no clear insight into the design of an "ideal" free improviser. Critical evaluations of this system by a wide variety of improvisers reveals a similarly broad range of opinions on how well it balances between engaging in supportive and oppositional behaviors. While the last two individuals discussed in detail above showed a preference for greater defiance and less reactivity in the system, many other individuals asked to critique the system did not agree with this assessment. In the end, data from this study does not indicate conclusively one way or another whether the system should be designed to be more supportive or more aggressive in its interactions with human improvisers.

However, as I have previously argued [14], commentary elicited in tests of this system have a value which goes far beyond simply refining the design of interactive music systems. Asking players to critique the playing of machines built to interact with human players like free improvisers elicits a discussion of what conduct is preferred in these interactions. In other words, the confrontation with a non-human musician brings improvisers to discuss the sense of *ethics* which they enact in how they listen and react (or not) to other performers.

Generally speaking, critical commentary on this system is useful for complicating any simple understanding of behaviors or dispositions such as "sensitivity," "supportive," "aggressive," or "defiant" as descriptors for the behavior of an improviser, whether human or machine. In the case of the improviser who found the system's behavior fickle, the system's behavior can be said to be both *too* sensitive and *not* sensitive *enough.* On the one hand, the system's inability to filter out or ignore what the human player was doing could be described as too sensitive or too responsive. On the other, this unfiltered hyper-reactivity is itself a musical behavior which reflects the system's failure to interpret the intentions and desires of the other player, a kind of interaction which could also be described as *a lack of sensitivity.*

5. CONCLUSION

The diversity of opinions on this system's behavior provides no clear answers for how this system ought to be re-designed. Similarly, this broad range of opinions fails to indicate with any finality how one might design an "ideal" improviser that might satisfy all tastes. While such inconclusivity may be frustrating for a designer simply looking for the best way to build a new system, analysis of the results of this study along these lines is rather short-sighted.

To mine commentary on how this system behaves for insights into further systems' development is to miss the tremendous opportunity it provides to empirically investigate the nature of social interactivity, whether through music or other expressive modes such as language, as specific forms of culture. Specifically, commentary on this system reflects a broad range of *notions of freedom* and ethics which guide how players engage in moment-to-moment decision-making in the course of their improvisatory interactions with other individuals. For those who desired greater sympathy from the system, their opinion reflects a general belief that the autonomy of one individual must be exercised in a manner such that it does not infringe upon the experience of liberty for others. Conversely, those who desired the system to demonstrate greater autonomy implicitly advocated a very different conceptualization of the relationship of freedom and ethics: the more that the system's behavior was uninfluenced and autonomous in relation to theirs, the more they experienced freedom themselves.

Acknowledgments

Field research for this project would not have been possible without financial support from the Mellon Foundation, Fulbright Journalism Fellowship (Germany), the Berlin Program for Advanced German and European Studies. Thanks CNMAT for support for publication of this paper, and to Adrian Freed, Benjamin Brinner and anonymous reviewers for their helpful comments.

6. REFERENCES

[1] G. E. Lewis, "Interacting with latter-day musical automata," Contemporary Music Review, vol. 18, pp. 99-112, 1999.

[2] G. E. Lewis, "Too Many Notes: Computers, Complexity and Culture in Voyager," Leonardo Music Journal, vol. 10, pp. 33-39, 2000.

[3] G. Assayag and S. Dubnov, "Using Factor Oracles for Machine Improvisation," Soft Computing, vol. 8, pp. 604-610, 2004.

[4] R. Banerji, "Maxine Banerji: The Mutually Beneficial Practices of Youth Development and Interactive Systems Development," eContact! Journal of the Canadian Electroacoustic Community, vol. 12, 2010.

[5] T. Blackwell and M. Young, "Self-organised music," Organised Sound, vol. 9, pp. 123-136, 2004.

[6] O. Bown, "Experiments in modular design for the creative composition of live algorithms," Computer Music Journal, vol. 35, pp. 73-85, 2011.

[7] B. Carey, "Designing for Cumulative Interactivity: The _derivations System," in Proceedings of the International Conference on New Interfaces for Musical Expression, Ann Arbor, Michigan, 2012.

[8] D. P. Casal and D. Morelli, "Remembering the future: applications of genetic co-evolution in music improvisation," in Proceedings of the European Conference on Artificial Life, 2007.

[9] N. M. Collins, "Towards autonomous agents for live computer music: Realtime machine listening and interactive music systems," Ph.D. Thesis, Faculty of Music, University of Cambridge, 2006.

[10] W. Hsu, "Using Timbre in a Computer-Based Improvisation System," presented at the Proceedings of the International Computer Music Conference, 2005.

[11] A. Linson, "Investigating the cognitive foundations of collaborative musical free improvisation: Experimental case studies using a novel application of the subsumption architecture," Ph.D. Thesis, Faculty of Mathematics, The Open University, 2014.

[12] M. Young, "NN music: improvising with a 'living'computer," in Computer music modeling and retrieval. Sense of sounds, R. Kronland-Martinet, S. Ystad, and K. Jensen, Eds., ed: Springer, 2008, pp. 337-350.

[13] D. Borgo, Sync or swarm: Improvising music in a complex age: Continuum, 2005.

[14] R. Banerji, "Maxine's Turing test–a player-program as co-ethnographer of socio-aesthetic interaction in improvised music," in Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE'12) Conference, 2012.

[15] T. Blackwell, O. Bown, and M. Young, "Live Algorithms: towards autonomous computer improvisers," in Computers and Creativity, ed: Springer, 2012, pp. 147-174.

[16] T. Jehan and B. Schoner, "An audio-driven perceptually meaningful timbre synthesizer," Proceedings of the International Computer Music Conference, Havana, Cuba, 2001.

[17] M. S. Puckette, T. Apel, and D. Zicarelli, "Real-time audio analysis tools for PD and MSP," in Proceedings of the International Computer Music Conference, 1998.

[18] V. Iyer, J. Bilmes, M. Wright, and D. Wessel, "A novel representation for rhythmic structure," in Proceedings of the 23rd International Computer Music Conference, 1997, pp. 97-100.

[19] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams, "The Timbre Toolbox: Extracting audio descriptors from musical signals," The Journal of the Acoustical Society of America, vol. 130, pp. 2902-2916, 2011.

[20] M. Malt and E. Jourdan, "Zsa. Descriptors: a library for real-time descriptors analysis," in Proceedings of Sound and Music Computing (SMC 2008), Berlin, Germany, 2008.

[21] D. Bailey, Improvisation: its nature and practice in music: Da Capo Press, [1980] 1993.

Bio-Sensing and Bio-Feedback Instruments
--- DoubleMyo, MuseOSC and MRTI2015 ---

Yoichi Nagashima
Shizuoka University of Art and Culture
nagasm@suac.ac.jp

ABSTRACT

This report is about new instruments applied by biological information sensing and biofeedback. There were three projects developed in 2015 - (1) a new EMG sensor "Myo" customized to be used as double sensors, (2) a new brain sensor "Muse" customized to be used by OSC, and (3) an originally developed "MRTI (Multi Rubbing Tactile Instrument)" with ten tactile sensors. The key concept is BioFeedback which has been receiving attention about the relation with emotion and interoception in neuroscience recently. The commercialized sensors "Myo" and "Muse" are useful for regular consumers. However, we cannot use them as new interfaces for musical expression because they have a number of problems and limitations. I have analyzed them, and developed them for interactive music. The "DoubleMyo" is developed with an original tool in order to use two "Myo" at the same time, in order to inhibit the "sleep mode" for live performance on stage, and in order to communicate via OSC. The "MuseOSC" is developed with an original tool in order to communicate via OSC, in order to receive four channels of the brain-wave, and 3-D vectors of the head. I have reported about the "MRTI2015" in past conferences, so I will introduce it briefly.

1. INTRODUCTION

As a composer of computer music, I have long been developing new musical instruments as a part of my composition[1]. The appearance of new sensor technology, interfaces, protocols and devices have led to new concepts in musical instruments and musical styles. I particular, biological sensors for EMG/EEG/ECG signals are very useful for musical application because the bio-information is tightly concerned with the human performer in a musical scene.

Inspired by "BioMuse" from Atau Tanaka[2] and research by R. Benjamin Knapp[3], I have developed five generations of EMG sensors from the 1990's, and developed many EMG instruments (called "MiniBioMuse" series) and methods of pattern recognition of performances[1]. The biggest advantage of EMG sensing is its short latency / fast response compared with other interfaces like switch, shock, pressure and CV sensors. At the 5th generation EMG sensor, I developed a XBee wireless interface, so the musical performance could be separated from the system. The freedom of not having cables is an important factor in live performances.

On the other hand, the private development of systems had a disadvantage in that the system is not as compact as mass-production. Therefore, I have recommended arranging or remodeling consumer products in developing new instruments. This is very good for education and hobby - arranging / remodeling everything (sketching) for original and customized use, and of course, in computer music and media arts.

Recently we can get many smart systems in the bio-sensing field - thanks to sketching (physical computing), 3-D printing and Open-source culture. Past bio-sensors were developed for medical use, so the systems were quite expensive. However, we can get smart, lightweight and usable bio-sensor systems now; the systems are not regulated for medical use, only for consumer / hobby use. In this paper, I will report two test cases of bio-sensors arranged / remodeled for computer music use.

2. MYO AND MYO-OSC

The Myo[4] (Figure 1) was supplied by Thalmic Co. in 2015. The "Myo armband" is constructed with eight blocks connected by a rubber connector, has eight channels EMG sensors and 3-D direction sensors, 3-D gyro sensors and 3-D acceleration sensors. The communication of Myo vs host PC is by Bluetooth, using the specialized interface "USB-Bluetooth dongle".

Figure 1. The "Myo" armband.

A specialized application "Myo armband manager" is supplied, and normal users can register the standard five poses to the application - fisting the palm, opening the palm, turning the palm to the left, turning the palm to the right and relaxing the palm. The specialized mapper in the "Myo armband manager" can assign these five poses into any keyboard codes, so normal users can control any commands with any other applications. For example, one pose changes the page of the presentation and another pose starts / stops the movie.

The standard Javascript tool "myo.js" for "Myo armband manager" can display all sensors information on the HTML screen in realtime (Figure 2). However, this Javascript interface "myo.js" using WebSockets cannot communicate with Max6.

Figure 2. Sensors data from "Myo" armband.

Next, I found the tool "myo-processing" (Figure 3 shows a screenshot). The standard sketch of Processing "myo-processing" can communicate with the "Myo armband manager", and displays all the sensor information.

Figure 3. Screenshot of the "myo-processing".

I have already experienced multi-process communication within the "Processing - Max6 - SuperCollider" system. I arranged the myo-processing sketch by embedding the OSC module, and succeeded in realizing "Myo - Processing - Max6" system, not only receiving Myo's 8+3+3+3 sensors data but also sending a "ping(vibration)" command from Max6 (Figure 4).

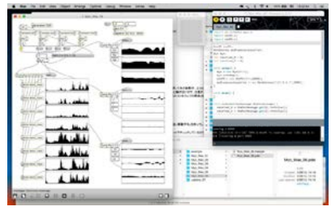


Figure 4. Screenshot of Myo-Processing-Max6.

After mastering the communication with Myo, I recalled that I have performed on many stages with my originally-developed EMG sensors (Figure 5). The important conditions for the live performance of computer music are - stability, long battery life, reproducibility of rehearsal, and system reliability. The "Myo" has powerful CPU/firmware, however the intelligent "sleep" function works against realtime performance. As is well known, the silent / still scene is an important part in music, but the "Myo" sleeps when the performer relaxes or is still on the stage.

The other request to the "Myo" is - to use both arms at the same time, like my past EMG sensors. When I added the second "Myo", the "Myo armband manager" could detect each Myo on the connecting window. However, there was no method to identify double "Myo"s with the Processing tools and the Java tools.

Figure 5. Performances with original EMG sensors.

3. THE DOUBLE MYO

In researching the Myo developers site, I found that the application "MyOSC" can deal with two "Myo"s individually. This application receives individual 3+3+3 sensors data from the double Myo, however, it can deal only with 3+3+3 data, and it cannot deal with the EMG data.

Hereon, I have decided to develop my original tool with the Xcode IDE, as the frontal attack. I analyzed the developers references deeply and tested many experimental prototypes. Eventually, I succeeded in communicating with "Myo" directly without the Processing-based tool. Figure 6 shows a screenshot - the Max6 can communicate with Myo via OSC by my original interface application.

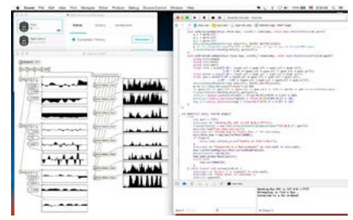


Figure 6. Screenshot of Myo-OSC-Max6.

Finally, I have completed developing an application that can communicate with double "Myo" individually and set the "non sleep" command (Figure 7). This meant that I can use "Myo"s in a computer music performance - with both my arms.

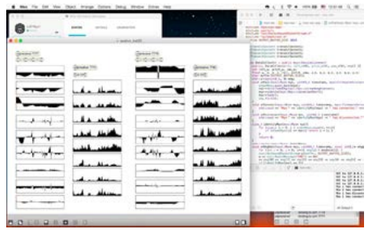


Figure 7. Screenshot of the "double Myo" test.

After that, I developed an experimental system to demonstrate the "doubleMyo" (Figure 8). With both eight channels EMG signals, this system generates 16 channels FM oscillator sound and realtime 3-D CG images (flying particles) with Open-GL. I have a plan to compose a new piece using this system.

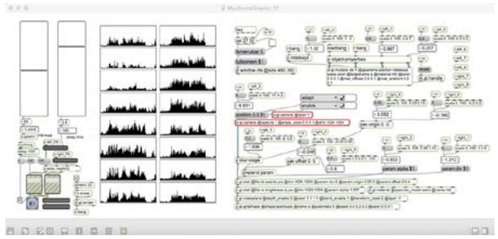


Figure 8. Screenshot of the DoubleMyo application.

4. THE MUSE

The brain sensing headband MUSE[5] (Figure 9) has been developed for relaxation and mental exercise. It was supplied by InteraXon Co. in 2015. MUSE has five-electrodes on the forehead, the second electrodes to ear-lobe, and a three-dimensional acceleration sensor. It is a compact lightweight inexpensive apparatus for transmitting biological information to the host via Bluetooth.

Figure 9. The "MUSE".

The regular user of MUSE uses a specially supported application for iPhone/iPad. The normal purpose of MUSE is mental exercise and mental health for amateurs - (MUSE is not authorized for medical usage). Figure 10 shows the usage style of MUSE and the result screen of the MUSE application. At first, people register their profile online. Every time after the brain exercises (relax), the users personal data is stored into the system, and people will try to realize "better relax on data" again with each training.

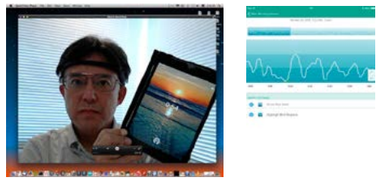


Figure 10. The "MUSE" usage and the application.

The "Myo" needs a special interface "USB dongle" to communicate via Bluetooth. On the other hand, the "MUSE" uses regular Bluetooth of the host system. I was not interested in the mental exercise, so I started to receive MUSE's Bluetooth information by Max's serial object (Figure 11).

Figure 11. Directly receiving the MUSE Bluetooth(1).

I tried to understand the complicated definitions and protocols of MUSE[6,7], and finally succeeded in receiving four channels of compressed brain-wave (alpha, beta, gamma and theta) data from MUSE (Figure 12).

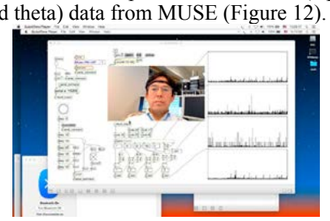


Figure 12 Directly receiving the MUSE Bluetooth(2).

5. MUSE-OSC

The disadvantage of directly receiving the MUSE via Bluetooth is dealing with high speed serial in the "Max". It is difficult to communicate with bidirectional protocol. By the way, the MUSE tool "MuseIO" supports the special protocol "like" OSC. The OSC is very familiar with "Max", so I researched the documents.

The standard "MuseIO" uses the TCP protocol which is not compatible with the OSC. I analyzed the documents and developed the special tool (Unix scripts) to set up the "MUSE" - as a UDP-based real "OSC" and notch filter option for power-line noise reduction (50Hz/60Hz). Figure 13 shows an experiment to change the filter parameter and to check the effect of noise reduction.

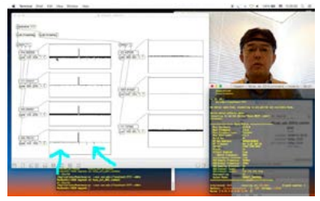


Figure 13 OSC test with the MUSE.

After the success of the OSC communication with the "MUSE", I found a big problem in using this system for musical performances. In the "mental exercise" mode, the MUSE application says "close your eyes, relax". After some minutes relaxing, the system starts. If I open my eyes and look at something, the system scolds me because so the EMG signals (around my eyes) interferes with the very weak brain signals. This means "MUSE cannot be used in a musical performance with ones eyes open". A contemplative performance (with eyes closed) can be beautiful of course. However, normally we cannot interact on the stage with other musicians in a musical session with our eyes closed.

Figure 14 shows the experiment of an AGC test. In the Max window, the vertical three graphs on the left side means the 3-D direction sensors, and the remaining nine graphs are the focus of the commentary. The left vertical four graphs are original input of the compressed brain-wave (alpha, beta, gamma and theta) from the "MUSE". With eyes closed, the brain-wave level is very small. Some of the big signals are the signals of the extraocular muscles of eye-blinks, and the signal of the facial muscles.

The center vertical four graphs are amplification to 10 times of the compressed brain-wave from the "MUSE". The brain-wave is well amplified to analyze the pattern, however the noise is overflowing the scale.

The right vertical four graphs are the AGC (automatic gain control) result. The amplification to 10 times of the compressed brain-wave from the "MUSE" is well amplified, and the noise signals are well compressed so as not to overflow. This algorithm is very simple (showed in the left side sub-patch).

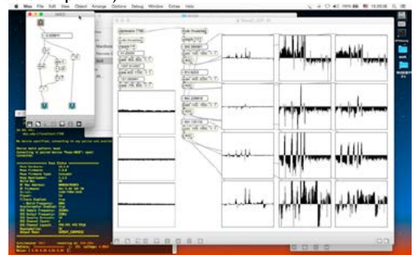


Figure 14 AGC test with the MUSE.

After this experiment, I decided that the "MUSE" should be used only as a sensor for extraocular muscles and facial muscles in musical performances. Because the time constant of the brain-wave is very big (slow reaction), this decision is effective for interactive live performances. I intend to compose a new piece using this system.

6. ADD THE MRTI2015

Working with the project "doubleMyo" and the project "MuseOSC", I developed a new instrument in 2015 featuring rubbing / tactile sensors. Because I have submitted a paper to another conference (NIME) about this, I will introduce it simply here.

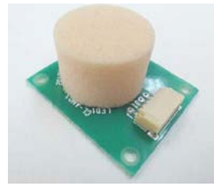


Figure 15 The PAW sensor.

The "RT corporation" in Japan released the "PAW sensor" in 2014. The "PAW sensor" (Figure 15) is a small PCB (size 21.5mm * 25.0mm, weight 1.5g) with a large cylinder of urethane foam on it. The output information of this sensor is four channel voltages which is time-shared conversion, which means the nuances of rubbing/touching the urethane foam with ones fingers.

My first impression was that I did want to use 10 "PAW sensors" with ten fingers. If all sensors are placed on the same plane like keyboard, the style of musical performance seems unnatural, because all finger tips must not move as on a piano or organ. A cube or mechanical shape is also unnatural for fingers/hands to grasp. Finally I found an egg-shaped plastic container (Figure 16). The experimental demonstrations are on YouTube[8-10].

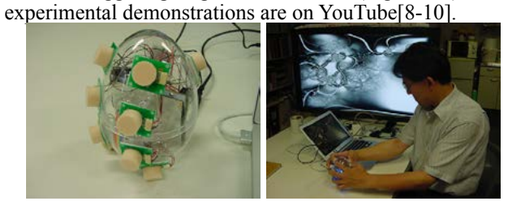


Figure 16 The new instrument "MRTI2015"(left) and its performance demo 8right).

After the three types of new instruments were developed, I aimed to mix up all sensor information, all sound synthesis parameters and all realtime graphic generation. Figure 17 shows the concept of the system, the environment is "Max7" and the communication is Bluetooth and USBserial.

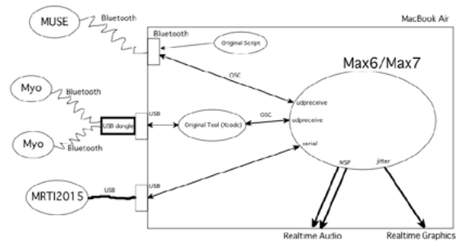


Figure 17 Mixture the three systems.

In the "Max" environments, I merged the three patches of "doubleMyo", "MuseOSC" and "MRTI2015" and tested, arranged and improved them. Figure 18 shows the testing process. I managed the OSC ports for "doubleMyo" and "MuseOSC", and merged the information using the same format/protocol for "MRTI2015" high-speed (115200) serial communication.

Figure 18 Mixture the three systems.

For the future tour, the "battery" issue is very important - battery life from full charge, irregular reset, auto-sleeping trouble and the calibrations. Figure 19 shows the battery check in the development experiments. The results are that the "Myo" works over 90 minutes continuously and the "MUSE" works over 3 hours after full charge, which is enough.

Figure 19 Battery check in the experiment.

7. DISCUSSION: THE BIOFEEDBACK AND MUSICAL PERFORMANCE

In this section, I will discuss the background of my research from the perspective of experimental psychology and brain science[3]. I have been researching bio-instruments and interactive multimedia art[11-17]. This is why I am interested in this field.

Damasio proposed the "Somatic Marker Hypothesis" in the area of brain science with the "as if loop" for the fast response in the brain (Figure 20)[18-22]. The "Somatic Marker Hypothesis" is pointed out to be the background of affective decision-making[23], or the background of interoception and emotions[24]. The interoception is a contrasting concept to the external senses (five senses). Each external sense has a specialized sensory organ. However, the interoception is organized from internal organs and the nervous system.

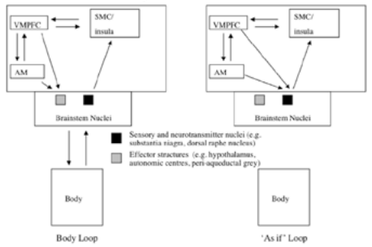


Figure 20 Somatic Marker Hypothesis [20].

As Figure 21 shows, Seth et al. proposed the interoception and biofeedback model as the background of the decision-making and feeling/emotion[25,26]. For example, the origin of exciting or dynamic emotions is from endocrine substances and hormones which are the result of human activity. The reaction time from this chemical object is long, but the "as if loop" works quickly as a short-cut in the brain. The differences between the result of the "as if loop" and the real result from the chemical response route are compared in realtime, and the prediction model is adjusted in real-time. With this bio-feedback mechanism, the "adjusted difference" occurs in emotion and the decision-making.

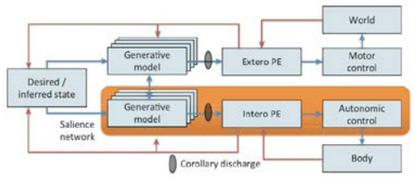


Figure 21 Seth's interoception and biofeedback model.

As Nacke et al. pointed out, bio-feedback is very important in the interaction design field[27]. There are many reports and papers on this topic, and I also reported on a EMG biofeedback game with gesture recognition system [28,29]. The subjects in the experiment do not know how to control/trim the muscle to replay the gesture with EMG sensors - this is of the interoception. However, most of the subjects can subtly control and realize the past-recorded gesture by unconscious trial and error. When the replay is succeeded by the bio-feedback graphical report, all the subjects feel happy/relaxed and positive emotions. This phenomenon will suggest great ideas to game-design, and to interactive live performance in computer music.

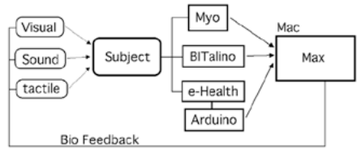


Figure 22 System design for future experiments.

Figure 22 shows the system block diagram of my future research in this field. We cannot detect the exact value of interoception because the value is chemical information or virtual in brain (in "as if loop"). So, the bio-feedback route to the subjects is well-known external senses channel - Visual, Sound and Tactile. However, the sensors to the subjects is bio-sensors - EMG, ECG, and EEG - Myo, MUSE, BITalino, e-Health and MRTI2015, etc. All sensing information is merged to the Max system and interpreted, and the visual / sound / tactile output is displayed into the subject to generate an affective response. This seems both kind of a game and kind of a mental relaxing exercise, and is a friendly interface in musical performance.

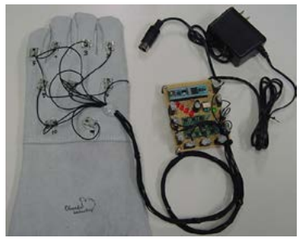


Figure 23 New tactile interface with linear actuators.

Figure 23 shows the newest experimental system of the tactile interface. I use ten pieces of "Linear Vibration Actuators" in system, controlling each vibration frequency with 32-bits resolution and high-speed response via MIDI[30], which I will report in the near future.

8. CONCLUSIONS

This report is about new instruments applied by biological information sensing and biofeedback. The perspective of experimental psychology and brain science are very interesting in considering new musical instruments and in creating new styles of music. Musical emotion is a very old theme, however, we can approach this theme now with the newest technology and ideas. I believe that computer music can open a new door in human emotion via new research.

9. REFERENCES

[1]Art & Science Laboratory. http://nagasm.org

[2]Atau Tanaka, BioMuse. http://www.ataut.net/site/BioMuse

[3]R. Benjamin Knapp. http://www.icat.vt.edu/users/r-benjamin-knapp

[4]Myo. https://www.thalmic.com/en/myo/

[5]MUSE. http://www.choosemuse.com/

[6]http://developer.choosemuse.com/protocols

[7]http://developer.choosemuse.com/protocols/bluetooth-packet-structure/compressed-eeg-packets

[8]http://www.youtube.com/watch?v=LF7KojKRP2Y

[9]http://www.youtube.com/watch?v=2SD84alrN1A

[10]http://www.youtube.com/watch?v=FM1Af3TyXNk

[11]Yoichi Nagashima, BioSensorFusion:New Interfaces for Interactive Multimedia Art, Proceedings of 1998 International Computer Music Conference, International Computer Music Association, 1998

[12]Yoichi Nagashima, Real-Time Interactive Performance with Computer Graphics and Computer Music, Proceedings of the 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Man-Machina Systems, International Federation of Automatic Control, 1998

[13]Yoichi Nagashima, Interactive Multi-Media Performance with Bio-Sensing and Bio-Feedback, Proceedings of International Conference on Audible Display, 2002

[14]Yoichi Nagashima, Interactive Multimedia Art with Biological Interfaces, Proceedings of 17th Congress of the International Association of Empirical Aesthetics, 2002

[15]Yoichi Nagashima, Bio-Sensing Systems and Bio-Feedback Systems for Interactive Media Arts, Proceedings of 3rd International Conference on New Interfaces for Musical Expression, 2003

[16]Yoichi Nagashima, Combined Force Display System of EMG Sensor for Interactive Performance, Proceedings of 2003 International Computer Music Conference, International Computer Music Association, 2003

[17]Yoichi Nagashima, Controlling Scanned Synthesis by Body Operation, Proceedings of the 18th International Congress on Acoustics, 2004

[18]Antonio R.Damasio. The Feeling of What Happens: Body and Emotion in the Making of Consciousness, Mariner Books,2000

[19]Antonio R.Damasio,. Looking for Spinoza: Joy, Sorrow, and the Feeling Brain, Harvest, 2003

[20]Antonio R.Damasio. Descartes' Error: Emotion, Reason, and the Human Brain, Penguin Books, 2005

[21]Antonio R.Damasio. Self Comes to Mind: Constructing the Conscious Brain, Pantheon, 2010

[22]B.D.Dunn, T. Dalgleish, A.D. Lawrence. The somatic marker hypothesis: A critical evaluation, Neuroscience and Biobehavioral Reviews 30 (2006) 239–271

[23]Y.Terasawa and S.Umeda. Psychological and neural mechanisms of interoception and emotions, Japanese Psychological Review, 2014,Vol.57, No.1, 49-66 (in Japanese)

[24]H.Ohira. Functional association of brain and body underlying affective decision-making, Japanese Psychological Review, 2014,Vol.57, No.1, 98-123 (in Japanese)

[25]Anil K. Seth. Interoceptive inference, emotion, and the embodied self. Trends of Cognitive Science, 17, 565-573. 2013

[26]L.F.Barrett and A.B.Satpute. Large-scale brain networks in affective and social neuroscience: towards an integrative functional architecture of the brain, Current Opinion in Neurobiology 2013, 23:1–12

[27]Nacke, L.E. et al. Biofeedback Game Design - Using Direct and Indirect Physiological Control to Enhance Game Interaction. Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems. 103-112. 2011

[28]Yoichi Nagashima. EMG Instruments as Controllers of Interoception --- for Healing Entertainment ---, Reports of Japanese Society for Music Perception and Cognition, 2015 (in Japanese)

[29]Yoichi Nagashima. A study of interoceptive entertainment with bio-feedback, Proceedings of Entertainment Computing 2015 (in Japanese)

[30]http://www.youtube.com/watch?v=7rvw_5Pshrs

FLUID CONTROL – MEDIA EVOLUTION IN WATER

Christoph Theiler, Renate Pittroff

wechselstrom (artist group)
Vienna, Austria

christoph@wechsel-strom.net, renate@wechsel-strom.net

Abstract

We have developed water based electronic elements which we built into electric circuits to control different parameters of electronic sound and video tools. As a result of our research we have constructed a complex controller whose main component is water. This tool makes it possible to control analog and software synthesizers as well as video software and other electronic devices, especially microcontroller based platforms like Arduino or Raspberry.

Keywords

controller, computer interface, water, electronic music, video, mass inertia, fluid, potentiometer, switch, fader

Introduction

Many traditional music instruments such as violins, guitars, timpani, pianos, and trumpets can give the musicians an immediate tactile response to their play. A strike on the timpani makes the mallets bounce back in a very specific manner, depending on the velocity, intensity, point, and angle of the beat. Plucking a guitar string, bowing a violin, sounding a trumpet or pushing a key on the piano not only requires overcoming a resistance but it also produces a kickback. On a piano for example, this kickback consists of the hammer falling back, an effect which the musician, upon touching the keys, can feel directly in his fingers. The nature and strength of this kickback response depend on both, the type of the action (plugging, beating, blowing, striking), and the strength, the sound quality, the pitch.

In electronic music the tactile feeling of the generated sound is absent. We cannot grab into the electric power and influence the sound quality with our hands in a direct manner. We cannot feel the swinging of an oscillating electric circuit consisting of transistors, resistors, and capacitors. Musicians have to play electronic instruments always in an indirect manner via interfaces.

These days the development of many industrially produced interfaces tends to avoid mechanical components as much as possible or to use only a minimum of mechanical parts. This leads to the fact that the input devices themselves do not create any music adequate resistance against the musician´s acting. Moving a fader or potentiometer from point zero up to half (50%) requires the same force as moving it from half to the top (100%). If this tool is used to influence the volume or the amount of distortion of a sound, one would wish for a fader whose sanding resistance increases according to the distance. Certain attempts have been made at finding a solution but the results have not yet gone beyond the status of a dummy, i.e. they are not actually included in the work circle of the sound production.

The best known example of such a development are the weighted keys of a keyboard. They are supposed to imitate the feel of a traditional piano but are not actually linked to the sound production. However, these particularities of the electronic sound generation do not imply a lack because the listener is rewarded with an immense amount of sound possibilities, a wealth that hardly exists in music produced with traditional instruments. On the other hand we have to admit that these particularities clearly influence the aesthetic perception of the work. Especially in the beginning of electronic music people used to describe the sound as very mechanical.

Fluid Control

The artist group „wechselstrom“ has made an attempt to develop the potential: A first approach consisted of producing the movement of sounds in space with an interface that gives the musician a physically tangible reference to his actions. These movements are normally regulated with a pan knob or a joystick. We equipped the interior of a closable plastic box with metal wires that took over the function of inputs and outputs of a mixer. These wires were isolated from each other, i.e. they hung free-floating inside the plastic box (Fig. 1).

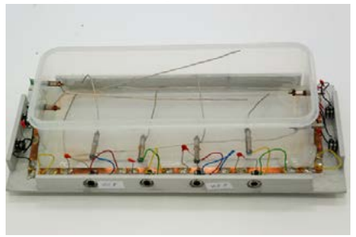


Fig. 1

The moment when the box was filled with (tap) water a complex structure of potentiometers was created mutually influencing each other. The wires took over the function of electrodes and the water served as a variable resistor. Measurements showed that the electrical resistance between two electrodes was between 15 - 50 kohms, depending on the immersion depth and the degree of wetting. These values are also used in normal potentiometers in electric circuits.

We have called this new instrument the „Fluid Control” box. It has been our goal to use Fluid Control as a matrix mixer which combines the functions of controllers, switches, faders, panning regulators, and joysticks in one hand. The movement of the water inside the box, the sloshing of the liquid reveals not just an audible image of the movement of sounds in space. Furthermore, the player / musician can bring his own body into a tactile relationship with the shifting weight of the water. The body and the instrument can now get into a resonant interaction. This process is similar to the rhythms of a sand- or rice-filled egg shaker which sound most lively when one succeeds to synchronize the movement of the grains with the swinging movements of the hand and arm.

In summer 2012 (during the festival Sound Barrier) we set up two Fluid Control boxes, two CD players, which resulted in a total of four mono tracks, and a 4-channel sound system. The four mono tracks coming from two CD players were launched into the input side of the first Fluid Control box mixed together with the appropriate proportion of water and sound levels on two tracks. This mixture was fed into the second Fluid Control box and distributed dynamically to the four channels of the sound system (Fig. 2).

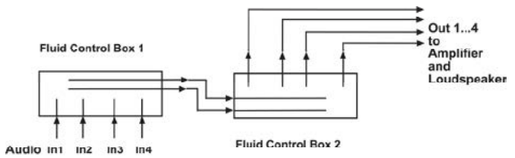


Fig. 2

Following the golden rule "current is current is current" the next step was to modulate not only audio signals but also to modulate control voltages generated in analog synthesizers. These electronic devices have the advantage of providing multiple physical inputs and outputs that can be plugged in directly. We showed this second setting for the first time on Sept 15th 2012 in the Jazzschmiede in Düsseldorf. We used the possibilities offered by Fluid Control for influencing the control current that was produced by an analog sequencer in order to drive an analog synthesizer (Fig. 3).

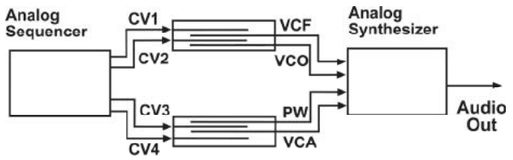


Fig. 3

As a result of our research we have created a tool which makes it possible to control electronic sounds within the dispositive of preselected sequencer and synthesizer setups in a very fast, dizzy, sophisticated, and sometimes chaotic way. Developing this tool we intended to make the change of the sound parameters in electronic music physically tangible. We also wanted to give the player a resistor / a weight into his hand which enables him to react in a more immediate and body conscious way to changes in sound beyond the scope of what controllers and interfaces like buttons, faders, rotary potentiometers, and touch screens can do.

As a the third we wanted to bring Fluid Control into the sphere of the digital wold of computers, software synthesizers and, as a follow up, of video or any other multimedia software. All well-known software synthesizers like MAX, pd, Reaktor etc. and most video/graphic software (MAX/jitter, Resolume) use and understand MIDI specification to control various parameters. We used a MIDI box which provided MIDI inputs and outputs and was connected via USB or FireWire to the computer on the other side at the same time. For the creation of a reliable MIDI data stream we took the +5 volt CV (Control Voltage) specification as an equivalent for the midi data value 0…127. We generated the corresponding data stream via a CV-to-MIDI converter. We modified the control voltage, which is often constructed with a single potentiometer, by adding the Fluid Control Box and by building it pre-, and/or post-fader or as a side channel into the electric circuit (Fig. 4).

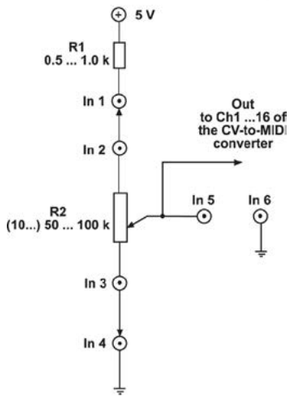


Fig. 4

"In1" and "In4" (socket symbol with arrow) are sockets with switching contacts, all other sockets are without switch. R1 is a resistor preventing a short circuit when sockets are connected in a wrong way (e.g. if you connect In1 to In6). The out goes to the input of one of the 16 channels provided by the CV-to-MIDI converter, which means that this circuit diagram was built 16 times (Fig. 5).

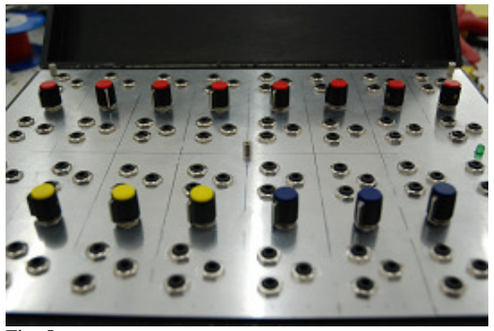


Fig. 5

Connections can be made between all sockets, even between sockets of different channels. However, only the following connections produce an effect: In1-In2, In1-In5, In2-In5, In3-In4, In3-In5, In4-In5 and In5-In6.

Fig.6, 7, and 8 show the basic connections. In Fig.6 two Fluid Control boxes are looped in. Together with R2 they build a voltage devider. When the slider of R2 is in the upper position the first Fluid Control box has more influence than box nr.2 and vice versa. When for instance the second box is plugged out the remaining box achieves the highest effect with the slider of R2 being in the upper position. When the slider is in the down position the box is inactive because the slider is connected to ground, therefore the output voltage is zero. In Fig.7 and Fig.8 the box achieves its highest efficiency when the slider is in the center position.

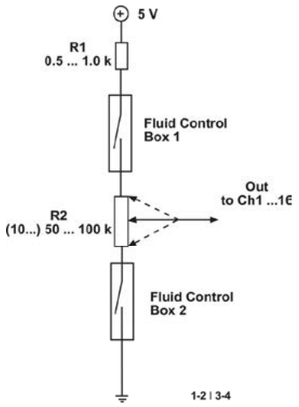


Fig. 6

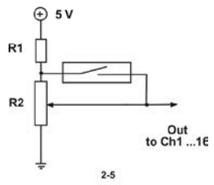


Fig. 7

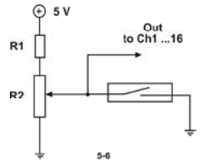


Fig. 8

Obviously, Fluid Control can be connected to any microcontroller or computer. In this case a MIDI-translation is not necessary, the circuits shown in Fig.4 – Fig.8 can be directly plugged into the analog inputs of the Arduino or Raspberry.

Film clips illustrating the operation of this instrument are available under the following Internet links:

How it works: (search for "Fluid Control Essenz")
https://www.youtube.com/watch?v=ed4JlMMNnyg
and "Fluid Control – The Installation"
https://www.youtube.com/watch?v=41uZi7bEdeI

wechselstrom

Christoph Theiler & Renate Pittroff

"wechselstrom" is a label owned by Renate Pittroff and Christoph Theiler. Based in Vienna, "wechselstrom" runs a so-called "offspace", which offers room for exhibitions, media activism and all art forms on the fringe of culture.

Selected works:

Piefkedenkmal – the construction of a monument for the musician Gottfried Piefke, who is also the namesake of the well-known Austrian derogatory name for Germans (2009 Gänserndorf)

Samenschleuder – a tool for environmentally conscious car driving (2009 Weinviertel, Lower Austria)

bm:dna – the government department for dna-analysis (2005 Vienna)

Tracker Dog - follow a (your) dog and track the route with a GPS, then print and distribute new walking maps (2008 Mostviertel, Lower Austria)

Community Game – a tool for distributing government grants using a mixed system of democratic vote and randomized control (2006 Vienna - distributing 125.000 Euro)

whispering bones – a theatre play asking for the whereabouts of A. Hitler´s bones (2004 Vienna, rta-wind-channel)

Reply - mailing action: resending Mozart´s begging letters under our own name to 270 people: to the 100 richest Germans and Austrians, to managers and artists of the classical music business, and all members of the Austrian government (2005/06 Vienna)

Re-Entry: Life in the Petri Dish-Opera for Oldenburg 2010

www.wechsel-strom.net, www.piefkedenkmal.at
www.samenschleuder.net, www.trackerdog.at

"MUCCA": an Integrated Educational Platform for Generative Artwork and Collaborative Workshops

Takayuki Hamano
Tokyo University of the Arts
takayuki.hamano@mail.com

Tsuyoshi Kawamura
TAJISOFT Dev.
kawamura@tajisoft.jp

Ryu Nakagawa
Nagoya City University
ingaz@mac.com

Kiyoshi Furukawa
Tokyo University of the Arts
furukawa@fa.geidai.ac.jp

ABSTRACT

Generative art is one way of creating art, and has been facilitated by digital technology. It has several areas of educational potential in terms of students being able to learn many kinds of artistic expression. However, existing environments have some limitations for introducing them in an educational setting. Thus, we have built an integrated educational platform for creating generative art and holding collaborative workshops. When designing it, we considered some aspects of learning processes and managing workshops. For the technical part of the platform, we developed environments for producing art and collaborating on projects, including a mobile application and a network system for holding workshops. Using this platform, we ran a workshop where participants made art by attending participatory concerts and interactive exhibitions. The response to this event demonstrated the efficiency of our platform, and pointed to the need for further investigation of learning experiences.

1. INTRODUCTION AND RELATED WORK

1.1 Generative Art and Education

Generative art uses an autonomous system that is employed in many creative domains such as visual art, architecture, and music. Although the core ideas of generative art have existed since old times, digital technology has advanced the process of making art. Regarding the history of generative art, MkCormack et al. [1] stated that computers and associated technological progress have brought new ideas and possibilities that were previously impossible or impractical to realize. Generative art has often been used as a learning tool in the context of education. Historically speaking, *Logo* [1] is one of the famous educational programming languages developed by Wally Feurzeig and Seymour Papert; it enabled students to produce graphics by programming the motion of a small visual robot called Turtle. *Processing* is another programming language, which is popular today for those who make media art. Pearson has collected diverse methods of generated graphic art in a book with *Processing* codes [2]. As for music, *Sonic Pi* [2] has recently become popular. Since global interest in education that involves information and communications technology (ICT) has been growing, demand is rising for learning environments that can facilitate self-expression through digital media. These software environments allow students to learn how to make a specific type of art by providing them with the knowledge to create it; previously, only experts could develop such art based on algorithmic models. These environments are also effective at enabling students to repeat trial and error by getting an immediate reaction from the system with respect to the operation. On the other hand, there are several points that such educational environments have in common. Firstly, it is still difficult for children below certain ages to learn how to generate art. Secondly, many types of software are technically designed for the solo user, even though they will be used in a classroom. In order to respond to the diverse demands of education, there is room for improvement.

[1] Logo Foundation
http://el.media.mit.edu/logo-foundation/

1.2 Motivation for Our Project

Based on what is described above, we would like to propose the effectiveness of bringing generative audio/visual art into art instruction by building an integrated educational platform that especially focuses on creating music. There was a predecessor to our project, an interactive, audiovisual piece called *Small fish* [3]. This work attempted to associate the visual movement of objects with the structure of music; the effect is similar to the actual physical world in that a ball that hits a wall produces a sound. This kind of cross-modal experience helps students learn about internal structures in musical expression. Our educational platform described in this paper expands on our previous work in order to achieve higher flexiblity and interactivity. The main purpose of this platform is to provide students with an opportunity to create work based on music theory, but with an intuitive operation. Furthermore, the platform aims to help develop the field of collaborative learning by utilizing the characteristics of music communication, similar to an ensemble performance. We expect that students

[2] Sonic Pi http://sonic-pi.net/

will eventually learn this new mode of artistic representation so that they will have a way to express themselves.

2. CONCEPTUAL DESIGN

Our educational platform consists of some technical bases that encompass both the individual creation process and the management of collaborative workshops. We have defined the basic concept of our educational platform (which aims to help people learn how music is algorithmically designed) as follows.

- The basic environment is a mobile application that runs on tablet devices. The application allows the user to produce audio/visual art while learning the ideas behind algorithmic design and musical expression, merely by touching a device and relying on an intuitive operation. The application allows for a high degree of expressivity in music, not only by allowing users to automatically generate music, but also to potentially realize a real-time performance.

- Aside from the mobile application, it is necessary to build an environment that enables students to cooperate with others, such that they can interact with each other and discuss their work. A function for students to share their work is a simple way of facilitating interaction among them. In terms of the musical experience, this function also provides students with an opportunity to form an ensemble, which leads to a collaborative music performance.

Based on these plans, we have developed an educational platform called MUCCA[3] (`http://mucca.town/`). This platform contains a mobile application for tablet devices, a network system for collaborative workshops, and a method for project management (in order to run workshops using those systems). We expect that students in elementary school, as well as older students, will use the platform, and thus participate in the creative experience.

3. SYSTEM DEVELOPMENT

3.1 Mobile Application

The first step to implementing the above-mentioned technical specifications is to develop MUCCA for iOS tablet devices. We used *Apache Cordova*[4] for the application platform, which produces a native mobile application based on HTML5 projects. As a result of being developed, the application has allowed the user to create music, with an intuitive graphical user interface. (Figure 1). The general procedure for creating music is as follows.

1. A visual object based on drawings and photographs is created, and placed on a two-dimensional canvas. A photograph can be taken with a built-in camera. Before placing an object on the canvas, an approximate outline of the object is automatically calculated for physical simulation.

[3] "MUCCA" originally stands for "Music and Communication Arts."
[4] `cordova.apache.org`

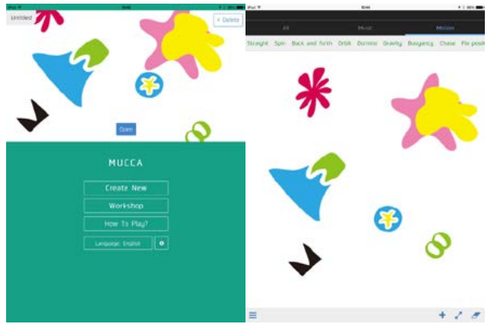


Figure 1. Screenshots of mobile application MUCCA.

Music	
Generation	Sound on motion, Sound on touching wall, Sound on contact, Remove sound
Scale	Major scale, Minor scale, Japanese In Scale, Japanese Yo Scale, Whole tone scale, Chromatic scale, Blues scale
Tempo	Fast, Normal, Slow
Range of Pitch	Wide, Normal, Narrow
Instrument	Celesta, Clarinet, Flute, Guitar, Harp, Piano, Trumpet, Viola, Recording, From DropBox
Motion	
Main	Straight, Spin, Back and forth, Orbit, Domino, Gravity, Buoyancy, Chase, Fix position

Table 1. List of major rules applicable to objects.

2. Rules for generating music to the object are assigned. The rules define the parameters, such as timing and tempo for producing sound, or the specific instruments involved. Sounds can be recorded with a built-in microphone and used as an instrument.

3. Motion is assigned to the object. Since a physical simulation engine is working on the canvas, the collision of objects is precisely captured, and the objects bounce in a different direction depending on their shape. If the user touches and swipes the object, it continuously moves across the canvas; its trajectory varies according to the assigned motion. The user can control the speed of an objects movement by swiping.

As described above, users can produce a musical structure and automatically generate a piece by creating a visual object on the canvas and assigning rules to it. The application allows the user to think about the characteristics of an objects shape, motions, and musical attributes. Table 1 displays the list of rules that apply to objects.

3.1.1 Mapping Musical Parameters

In the application, the internal musical data generated by the user's manipulation is MIDI-like data. This means that the data contains information about musical notes; each note has an instrument channel, a pitch, a duration, and amplitude. Every factor is controlled on the user interface, such as the shape and motion of objects, or rules associated

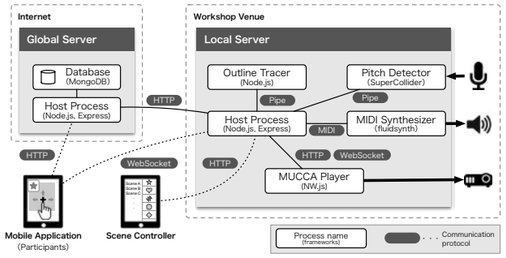


Figure 2. Internal architecture of the workshop system.

with them, which affect how note data is generated. The timing of when notes are played can be selected from the rules in the category *Generation*. If the rule item *Sound on motion* is selected, the object can continuously make a sound while it is moving. The basic pitch is determined by the vertical position of the objects on the canvas, which means an object moves to the top of the screen, and the pitch changes from lower to higher. In addition, the pitch is modified based on the rules of the musical scale. In the case of a recorded audio sound, the user can change the pitch by adjusting the playback speed rate. The user can modify the time range for playing a sound back.

3.2 Network System for Workshops

The next step is to develop a workshop system that becomes the foundation for communication. We have determined the requirements of this workshop system as follows.

- The system allows users to submit created objects from a tablet device. In the workshop venue, submitted objects are shared on a screen, and users can make music as they would using the mobile application. Shared objects can be controlled in real time from any kind of tablet device.

- In order to create a musical piece via shared objects, the system allows users to compose musical scenes based on selected objects, and to form sequences of multiple scenes.

- Collaborating with acoustic musicians is a valuable experience for students during a workshop. It would be ideal if the system mediated interactions between students and musicians.

Considering all the requirements above, we have designed the internal architecture of the workshop system, as shown in the Figure 2. The workshop system comprises both local servers and a global one. The local servers are placed in each venue where a workshop is held, while the global server is placed on the Internet.

3.2.1 Local Server

The local server can display submitted objects and play music, using them in the same way as with the mobile application. This means that the participants pieces are interactively projected onto a large screen at the venue. Using the WebSocket communication protocol, this server also accepts control signals for objects projected from tablet devices. This type of server was developed based on the programming environment, *Node.js*[5] and a web application framework called *Express*[6].

For a concert, musical scenes are composed with objects created by participants. For this purpose, we built a scene controller in which facilitators can reorganize submitted objects and compose them as several scenes.

During the concert, the pitch of sounds played by acoustic musicians are analyzed in real-time with *SuperCollider*, and mapped to align with objects displayed on the screen. For example, when the system detects the pitch of the C4 sound, objects align horizontally, and objects scatter outside when the A3 sound is produced.

3.2.2 Global Server

The global server is placed on the Internet; it mainly manages submitted objects and authenticates local servers. It always accepts submissions from the mobile application, and relays information to the local server regarding the venue that the user specifies. Since we assume that workshops will be held simultaneously in many places, it makes sense for the global server to manage information about local servers.

4. CONDUCTING A WORKSHOP EVENT

To conduct a pilot test for our educational platform, we held three kinds of workshops for young students over two days in summer 2015 as a part of a series of events that inaugurated the cultural institution *Gifu Media Cosmos* in Gifu city, Japan. The first workshop was for creating art, the second one consisted of participatory concerts, and the third consisted of interactive exhibitions. These three workshops related to each other.

4.1 Workshops for Art Creation

Regarding the workshop for creating art, participants used MUCCA with support from facilitators. Around 15 participants from local elementary and secondary schools attended each session, and the participants were divided into 3 working groups. Each session was an hour-and-a-half long and comprised of two parts. In the first part, participants learned how to use the mobile application, and repeated trial and error when creating their own works; meanwhile, they discussed the process with the members of each group. In the second half, the workshops ended, the participants also enjoyed the improvisation ensemble by using a real-time control feature for tablet devices.

4.2 Participatory Concerts

The participatory concerts were occasions for presenting the creative results of the workshops, where participants attempted to produce musical pieces based on their collaboration with acoustic musicians. Prior to these concerts, we used the scene controller to compose several musical scenes from materials created by the participants. The participants played music using the controls on the application. In some musical scenes, acoustic musicians provided

[5] `https://nodejs.org/`
[6] `http://expressjs.com/`

Figure 3. Photograph of the participatory concert. The saxophone player is controlling objects created by participants.

accompaniment with instruments such as the accordion, the tenor sax, and the piano. Their sounds were partly applied in real-time to arrange objects on the screen, for the purpose of creating interactions between the participants and the musicians (figure 3).

4.3 Interactive Exhibitions

Visitors could take part in the interactive exhibitions at any time. Submitted materials continuously appeared one by one on the screen at the venue. Participants were able to submit materials from both within and outside the venue, where they were allowed to use their own smartphones to control objects on the screen in order to create a musical performance.

4.4 Reaction from Participants

Participants were fairly satisfied with the event. After every workshop session and concert, we received many comments from participants. There were three cases when they reported feeling pleasure: (1) Learning about the ideas and use of the application, as well as how it worked intuitively, (2) Interacting with other participants through the workshop system, and (3) Showing off their own work to friends or family and discussing it. It was very impressive that participants actively had debates in order to interpret each others pieces and viewpoints.

5. DISCUSSIONS

5.1 Improvement of Learning Experience

Our educational platform worked successfully in terms of the technical side throughout the workshop, but there are some points to be considered in relation to the participants' experience. Although the students seemed to learn how to operate the application very quickly so that they actively designed their work by themselves, we wonder if the workshop covered all types of learning. During the phase of individual creation, many of them were able to comprehend the relationship between visual factors and music. However, apparently only a few participants were able to assemble musical ideas by using an aesthetic judgement of music. Facilitators also played an important role in terms of guiding the participants toward their interests in art expression. We also need to improve the balance of time allocation for a better learning experience.

On the other hand, in terms of the collaborative music performance, it was very pleasant to observe numerous situations where participants discussed many ideas about a work they had created during the music ensemble, when they jointly produced music via the devices. We noticed that they often talked to each other about the characteristics of the music they had created together. We assume that collaborating directed the participants attention toward various perspectives.

5.2 Future Works

For the next step of this project, we expect that students will accumulate knowledge about generative art by participating in workshops. For this purpose, it is necessary to conduct our activities in various places and to open more workshops. Our platform is designed so that the workshop system can function simultaneously in multiple locations. We are currently planning to deploy the workshop system as a package, so that any educational institution can easily hold workshops independently. Furthermore, we wish to allow participants to share their work online at any time. Social media can also be utilized to develop an online posting forum to show the students works using MUCCA.

6. CONCLUSIONS

In this paper, we described the implementation and practice of our integrated educational platform, MUCCA. MUCCA is based on generative audio/visual art, and focuses on both the technical development and management of a workshop. The development of this platform was successful in terms of involving participants and their pieces; however, we need to further examine students learning experiences. We believe that MUCCA will become a sufficient platform for accumulating knowledge based on educational experiments that help people use generative art to express themselves creatively.

Acknowledgments

The author would like to thank Masao Tsutsumi (a staff member of the Gifu city government), Yukiko Nishii (the accordion player), Jo Miura (the saxophone player), Sayumi Higo (the visual designer), and all other members who were involved in the workshop.

7. REFERENCES

[1] J. McCormack, O. Bown, A. Dorin, J. McCabe, G. Monro, and M. Whitelaw, "Ten Questions Concerning Generative Computer Art," *Leonardo*, vol. 47, no. 2, pp. 135–141, Mar. 2014.

[2] M. Pearson, *Generative Art: A Practical Guide Using Processing*, pap/psc ed. Manning Pubns Co, 7 2011. [Online]. Available: http://amazon.co.jp/o/ASIN/1935182625/

[3] V. Grassmuck and H. Staff, *Zkm Digital Art Edition 3: Kiyoshi Furukawa, Masaki Fujihata and Wolfgang Münch*. Hatje Cantz Verlag, 2005.

Electro Contra: Innovation for Tradition

Benjamin D. Smith
Department of Music and Arts Technology, School of Engineering and Technology,
Indiana University-Purdue University Indianapolis
bds6@iupui.edu

ABSTRACT

Technological interventions in American traditional fiddle and dance music are presented and specific design and development problems are considered. As folk dance communities and events explore the notion of incorporating modern electronic dance music into the experience certain inherent problems are exposed. Maintaining strict musical forms that are required for the traditional choreography, maintaining the fluidity and control of live bands, and interacting with the other performers require new software tools. Initial solutions developed in Ableton Live are described and show a successful method of solving these challenges.

1. INTRODUCTION

Traditional aural music practices around the world evolve and maintain currency with the incorporation of new musical instruments and technologies. In the twentieth century steel strings for guitars and violins, the advent of amplification and electric instruments, and increased manufacture and access to instruments had transformative impacts on music around the world. New genres grew out of the new technologies, such as Jazz and Rock and Roll, exploding in dance halls and on concert stages alike. Amplification is now a ubiquitous aspect of dance music performance in nearly every genre, from social and couples folk dancing to swing to electronic dance music (EDM). Today, computers present an immense domain of musical possibilities and their incorporation as a performance tool in traditional folk music, alongside fiddles and banjos, is already underway.

Performing 'traditional music' electronically, on a technical level, presents many challenges to the electronic musician using currently available software tools. Most folk dance choreography fits strict musical forms and any musical deviations will disrupt the dancers and stop the dance. The music has to start and line up with the figures of the specific dance, requiring the musician to synchronize the phrasing with choreography. Further, the music is expected to dynamically respond to the dancers through texture changes and growth of a song, facilitating energetic and emotional experiences.

Based on these challenges several new software tools (plug-ins for Ableton's Live Suite) have been designed, developed, and evaluated. The specific goals and problems, primarily centering on phrasing and maintaining phrase alignment, lead to the implementation of three tools for performance use. These provide a relative beat jump, an absolute beat jump, and an automatic clip synchronization tool. Use in a series of performances and dance events show that these are effective in practice, but present challenges of their own and a need for further design and development.

2. CONTRA DANCE

American contra dance is a vibrant living tradition of dancing and music performance that has been steadily growing in popularity since the 1970s. Involving instruments, music, and choreography derived from eighteenth century practices in the British Isles, contra dance now has active communities across North America, Europe, and Australia. The current form of contra dance was first seen in the U.S. in the 1780s [3], and after disappearing from practice in the following century was reborn during the folk revival in the United States in the 1970s [8]. While the closely related forms of English, Scottish, and Irish dance followed the same trajectory they have become historically oriented practices, privileging traditional choreography and costumes. Uniquely, contra dance actively supports regional and individual variation, new choreography, and experimentation with the forms and music [4].

The structure of contra dance employs two lines of dancers (the designation "contra" refers to this opposition of lines), who progressively move along the lines to dance with other individuals. The choreography typically involves each sub-set of four dancers (two couples) executing a series of steps in unison that take up the 64 beats of the written dance [1]. All the dancers execute each figure in the dance concurrently and a series of 4-8 figures typically comprises a "dance," which is then repeated 12-20 times along with live musical accompaniment.

The vast majority of the choreography is set to a binary musical form of AABB, wherein each section is 16 beats long. The music is performed live and is historically rooted in the traditional music of the British Isles (Irish and Scottish 'fiddle' tunes). The meter is most commonly 2/2 or 6/8, and is strongly phrased to indicate the 8 bar sections, which dancers rely on for structural cues and to "keep them on track" [3]. Dance tempo does not vary widely, and is conventionally in the 115-125 beats-per-minute range [4].

The notion of "tradition" is integral to contemporary contra dancing, and the ideals of a non-commercial 'folk'

community and 'traditional' Americanness are primary components in drawing many to the group [8]. As such these values are felt strongly amongst the community and guide many aspects of direction and organization locally and nationally. Musically, these ideals privilege 'traditional folk' acoustic instruments (such as the fiddle, piano, banjo, and acoustic guitar), and tunes in strict musical forms (e.g. 2/2 metered Reels and Hoedowns; 6/8 metered Jigs and Marches).

However the authenticity of the 'tradition,' in terms of longevity of customs and practices, is largely a chimera [8]. While some smaller communities in the North Eastern U.S. maintain a closer aural, generational link to the ancestral dance forms [9], for modern urban contra dance the authenticity of the musical tradition, in terms of repertoire and performance practices passed down aurally from generation to generation, is non-existent. The community of dancers is intentional and associational, rather than based on ethnic, religious, or locational alignment [3].

The upholding of tradition creates friction with the living practice aspect of contra dance, leading many contemporary musical groups to both retain traditional instrumentation while experimenting with a diversity of genres and sounds. One of the most popular notional contra dance bands today, The Great Bear Trio [5], is lead by an electric guitar and regularly features arrangements of Top 40 radio songs. Another extremely popular band, Perpetual eMotion, used looping technology and extensive electronic effects applied to the fiddle and guitar to create live EDM-styled dance music [6].

The first noted examples of contra dancing to non-traditional pre-recorded music at mainstream contra dance events is thought to have occurred in the early 2000s in the Boston area [6]. This lead to alternative dances colloquially termed "techno contras" [2], being staged across the U.S. today. Many self-styled DJs use mixes of EDM, pop, world beat, and fusion music to stage these events at festivals annually. Almost all of these performers premix compilations of songs by other artists and play these tracks in a fixed fashion to accompany the dance. These DJs have further explored changing the nature of the event from the conventional series of approximately 10 minute dances interleaved with short breaks to more continuous sequences of dances (some reportedly stringing dances together for as long as 90 minutes without pause).

The desire to incorporate electronic dance music in contra dance events appears to be based on fostering intense emotional experiences [7] and perceived "altered states". Contra dance already creates these experiences for many through the highly repetitious dance forms and musical tunes, akin to a group recitation of a mantra [3]. Likewise, EDM is known for supporting similar experiences through looping, and iconic production techniques such as the "build-up" and "drop" [7]. The receptiveness of the otherwise traditionally oriented contra dance community to EDM type music may be based on this affinity for altered state experiences, allowing for this seemingly radical influx of distinctly non-traditional music.

Performing live, interactive electronic music for contra dances is currently being attempted by a few national acts, notably Buddy System, DJ D.R. Shadow, DJ Squeeze, and Phase X. These artists use a combination of DJ software, controllers, electronic and amplified acoustic instruments and effects. The work described herein is based on the experiences and findings of members of these groups.

3. MUSICAL STRUCTURE

The primary problems faced by live electronic music in the contra dance context stem from the strict requirements of the phrase structure and the need to aurally cue and indicate the repetitions in the form. The binary pattern of AABB, as well as the continual recycling of the whole form (over each 8-10 minute dance), are expected and relied on by the dancers [3]. This stands in contrast to the typical pop music song form of AABA and EDM forms which focus on continuity and minimalist trance-like repetition. Further, pop songs commonly deviate from 32 bar forms to include a bridge section or other variations, which precludes their use in this context.

The electronic performer can create the form by discarding loops and playing everything live using controllers and MIDI interfaces (i.e. treating their setup like an acoustic instrument and 'playing all the notes'). However this denies the hallmark sounds, sampled loops, and operating principles EDM is based on. The opposite approach seen above, of acoustic musicians playing contemporary pop songs for dances, merely appropriates the content of one genre and transposes it to another, rather than exploiting the potential of fully blending the genres.

Groups providing live music for contra dance must additionally be able to recover from errors enacted by the caller or dancers. While not common, either the caller may mistakenly call a figure or the dancers may forget and cause the dance to get out of sync with the music or come to a stop. It is imperative that the musicians are able to either resynchronize with the dance (by adding a few beats or skipping ahead in the song), or quickly reset and recover by starting over.

An additional problem arises solely at the commencement of each dance where the musician must either cue the start of the choreography or align with the caller/leader. Conventional acoustic contra dance bands start each dance in one of two ways: either by playing a short four beat introduction to indicate the start of the dance to the dancers, or by playing a repetitious musical pattern in the tempo of the dance and allowing the caller to time the figures to the music. In this later case once the dancers are all in motion the musicians will seamlessly transition to their full tune/song/arrangement.

Ableton Live is a preferred software solution for many live electronic musicians playing on the contra dance stage due to its flexibility and interactivity (see Fig. 1). The ability to play loops, clips, and songs dynamically and apply further manipulations is the basis for these performers. However the challenges of phrase alignment in this environment are seen as cumbersome and constraining to expressive performance. For example, if the user wants to change material in the middle of the 32-bar form there is no easy way to quickly trigger new loops and cross-fade or cut the old ones while still ensuring adherence to the dance structure. If the user accidently triggers clips or sections at the wrong time there is no way to recover without impact to the musical form.

Figure 1. Ableton Live Set used for contra dances (image courtesy Julie Valimont) showing density of musical tracks and clips.

4. NEW DEVELOPMENTS

Based on discussions with performing musicians three Live "devices" (plug-ins or utilities in Live's parlance) were proposed, developed, and tested. The overall goal is to ensure enforcement of the phrase structure, freeing the musician to focus on musical choices, texture and dynamic direction. The developed assistive utilities are:

1) Song jump device that instantly skips the entire session (all playing clips and events) to a specified bar and beat, or by a relative number of beats.
2) Track jump device that skips a single track to a specified bar and beat, or by a relative number of beats.
3) Clip synchronization device that maintains phrase alignment between a slave track and a master track (or the master clock).

All of these devices were built using Max For Live (M4L), working extensively through the Live API (in Max 7.2.2). This allowed easy modification during the prototyping stage as well as cross-platform distribution. The devices were used in performances during development, generating bug lists and feature requests stemming from real-world application.

The song jumping device (see Fig. 2) gives the player the ability to skip the song forward and backwards by single beats, assisting alignment with the dance if the music is out of sync, as well as jumping by whole sections to extend or shorten a song. This is analogous to a DJ moving the needle on a record, skipping the song to a new point in time. Ableton Live employs a model where each loop or clip is essentially an individual record with its own needle, and jumping the song causes all the clips to jump synchronously. The Live API exposes access to the master clock time ("current_song_time") which is set in the M4L device (through the "jump_by" function) when the user enters a new absolute or relative jump time.

The track jumping device performs similarly but only acts on a single track at a time, serving artistic effects and affording the alignment of different clips and loops. This uses the Live API "playing_position" property of a specific clip.

Figure 2. User interface for Song Jump device.

The clip synchronization device forces any track to stay aligned with either another track or the master clock. In Live the user can configure a quantization rate for clip launching, which causes clips to delay commencement to match a certain phrase length. That is, if the quantization rate is set at 2 bars clips will start playing when the master song clock is at even bar numbers regardless of when the user presses the clip launch button (see Fig. 3, showing misalignment resulting from the user triggering clips around the phrase point). While this effectively enforces clip alignment dynamically, longer phrase lengths (such as the 8 or 32 bar phrases in contra dances) present challenges and this quantization limits performer spontaneity. If the user triggers a clip one beat after the 8 bar quantization point the clip will wait 7 bars before playing (see Fig. 4). This limits the performer's ability to improvisationally mix the music and dynamically trigger new clips. The new clip sync device allows the user to turn off the global quantization, allowing any clip to launch at any time, and the device ensures phrase alignment (see Fig. 5 where clips start playing in the middle of their loop). As each clip is launched the device skips it to play from the point that aligns it with the configured phrase length.

Figure 3. Loops with quantization at 2 bars, long loops enter out of phase with 8 bar phrases.

Figure 4. Loops with quantization at 8 bars, aligned correctly with phrases, but limited flexibility.

Figure 5. Loops with no quantization and Clip Sync device. Loops can start in the middle with guaranteed phrase alignment.

In this device the phrase length can be set independently for each clip by the player (commonly 8 or 32 bars). For each given time point (t) the audio sample to play (X) is calculated from the time point of the master track (t_{master}) folded by the length of the phrase in samples (based on the user set length in beats P, the tempo T, and the

sample rate of the audio engine *sr*) and the length of the slave track's audio loop (*L*, in samples):

$$X_t = t_{master} \bmod PTsr \bmod L \qquad (1)$$

5. REFLECTIONS

All of these devices have been used in over a dozen performances, each lasting 1 to 3 hours, and have proven to be stable in practical use. The song jumping device was intended to solve the problem of transparent alignment with the start of the dance. In theory the musician would build a looped groove for the caller to teach the dance, and at the point the dancers have begun the choreography the musician would align and cross fade to their song tracks. To enact this alignment the musician would launch their song tracks at any point and at the moment the dance reaches the start of the choreography (the beginning of the first A section) the musician pushes the "jump to start" button, causing the entire song to jump to beat 0 and be aligned with the dance.

However, this operation in Live causes any clips that were launched after beat 0 to be turned off. Practically this results in everything stopping at the critical moment when the musician is aligning their song with the start of the dance. Thus this functionality does not work as intended. However the device enables smaller relative jumps by a beat or a bar, moving the playback for all playing clips simultaneously, and has proven useful as an error correcting measure if the dancers get out of sync with the music.

The track jump device does not suffer from this problem, since jumping a track to beat 0 still allows the track to keep playing. Thus this device solves the previous problem, of restarting the track when the dancers reach the start of the dance. It does not, however, allow many tracks to be moved simultaneously, but appears to be adequate for initial use (typically users start their arrangement for a dance with a single track, which this device enables, and then build from there). In combination with the third device the track jump solution has proven effective in quickly aligning the entire set.

As an error recovery tool, especially to realign after the caller or dancers make a mistake, the track jump device has proven highly successful. As long as the musician knows where the start of the first A section is in the dance they can use the track jump device to immediately jump to that point in the music to coincide with the dancers. This is a critical ability for the live music.

The clip alignment device appears to be the most transformative of the three utilities. This functionality allows a musician to start a musical loop or sample at any time point and ensure that it remains sample locked to a master track (or master clock). In practice this gives the musician a lot of freedom to start musical material without worrying about where in the form structure they are. Previously the musician had to remember the length of each sample clip they had loaded into their set and then trigger it precisely to align with the dance form. While Ableton provides a quantization method to ensure clips only start at certain points this prevents dynamic interleaving of new clips at a finer granularity. If the musician wants to start a new percussion line in the middle of the quantization length, or start in the middle of the percussion loop, it is now possible.

6. CONCLUSIONS

While these new devices successfully assist the live electronic musician in performing for contra dance events additional tools will be needed to support artistic creativity in performance. Several specific problems were teased out and addressed with new software tools that have been field-tested and are in current use by performing artists. Further, these devices may be useful to Ableton Live users generally, beyond the domain of folk dance music.

The interviewed musicians, all of whom have extensive experience as acoustic performers, continue to seek flexible ways of dynamically creating their music and interacting on the dance stage. This confluence of 'traditional' folk dance and electronic dance music is attracting musicians and dancers alike to events around North America and promises to continue serving as a locus for experimentation and growth. As new artists bring new approaches and new technology to the dance stage, new practices, instruments, and tools will be discovered and incorporated into these evolving traditions.

7. REFERENCES

[1] Dart, Mary McNab. 1992. *Contra Dance Choreography: A Reflection of Social Change* (Doctoral Dissertation: Indiana University).

[2] Foster, Brad. 2010. "Tradition and Change." *CDSS Blog*. Accessed February 23, 2016. http://blog.cdss.org/2010/12/tradition-and-change/.

[3] Hast, Dorothea E. 1993. "Performance, Transformation, and Community: Contra Dance in New England." *Dance Research Journal* 25 (01): 21–32.

[4] Kaufman, Jeff. 2008. "Dialectical Variation in Contra Dance." Master, Swarthmore College.

[5] ———. 2016. "Festival Stats 2015." Accessed February 23. http://www.jefftk.com/p/festival-stats-2015.

[6] Krogh-Grabbe, Alex. 2011. "Crossover Contra Dancing: A Recent History." *CDSS Blog*. Accessed February 23, 2016. http://blog.cdss.org/2011/06/crossover-contra-dancing-a-recent-history/.

[7] Ragnhi, Torvanger Solberg. 2014. "'Waiting for the Bass to Drop': Correlations Between Intense Emotional Experiences and Production Techniques in Build-up and Drop Sections of Electronic Dance Music." *Dancecult: Journal of Electronic Dance Music Culture* 6 (1): 61–82.

[8] Turino, Thomas. 2008. *Music as Social Life: The Politics of Participation*. University of Chicago Press.

[9] Young, Kathryn E. 2011. "Living Culture Embodied: Constructing Meaning in the Contra Dance Community." (Master Thesis, Denver, CO: University of Denver).

LR.step, an Algorithmic Drum Sequencer

Morgan Jenks
Texas A&M University
morgan.m.jenks@gmail.com

ABSTRACT

This paper presents a new algorithmic drum sequencer, LR.step. This sequencer is based on Clarence Barlow's Indispensability algorithm, and builds upon previous work with this algorithm, introducing several novel features. LR.step differs from previous implementations of the indispensability algorithm in that it features a method for calculating arbitrary subdivisions of the beat, such as 14th note triplets, as well as two new processes for generating syncopation. Details of the software and possibilities for future work are given.

1. INTRODUCTION

Music software design has become widely accessible with the advent of the Internet and online communities of practice. Vast collective knowledge enables creatives to design instruments that suit their own needs and ideals. LR.step exists as a result of a personal performance practice at the intersection of consumer electronics, longstanding research into music theory, the Max/MSP community, and my own stylistic interests, informed by ready access to various experimental beat makers such as Autechre[1].

1.1 The Algorithm

Many algorithmic sequencer techniques have been developed, including euclidean approaches, stochastic approaches, cellular automata, and genetic approaches [2, 3]. Among all of these, the indispensability algorithm, developed by Clarence Barlow in 1978, stands out as an interesting balance of musicality and flexibility [4].

The indispensability algorithm, to summarize, sorts all steps in a sequence, given a number of measures in a particular time signature and a subdivision of the meter as the step size, ranking the steps by their importance or 'indispensability' to the stable perceptibility of the meter. For example, a single 4/4 measure of 8th notes will have downbeats 1 and then 3 as the most important pulses, followed by 4, 2 and then the upbeats (emphasizing the antecedent to beat 1).

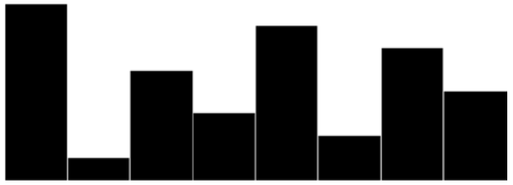


Figure 1. The indispensability set for one measure of 4/4 in 8th notes

The indispensability algorithm is a state machine much like cellular automata or euclidean rhythms. However whereas the latter two output sequences of Boolean values, indispensability sets provide a rich hierarchy for all possible pulses in a sequence and create conventional metrical emphasis, even in complex time signatures. The indispensability algorithm reveals connections between rhythm and harmony, and outputs patterns strikingly similar to traditional musics, for example, Franconian dance pieces [5]. Composer Georg Hajdu has ported the algorithm to Max/MSP and used it to assist in organizing a 19-tone equal temperament recorder piece among other things [5].

Barlow's original implementation of the formula took the form of his all-in-one procedural composition system, Autobusk [4]. Another implementation of the indispensability algorithm by Sioros, Guedes and the Kinetic Controller Driven Adaptive Music Systems Project at the University of Texas, Austin, uses the indispensability set as a probability table and features real time control of meter, subdivisions, and probability weight [6]. As Eldridge advocates regarding musical generativity, the indispensability algorithm is not lifted from another scientific context such as flocking simulation, but was devised specifically from harmonic and metrical principles [7].

2. SEQUENCER DETAILS

LR.step, in its current form, is a Max/MSP patch that syncs with Ableton Live via the ReWire protocol. In contrast to the Autobusk and Kinetic rhythm generators, LR.step is not stochastic. It is fully determinate and will output a consistent and static pattern for a given combination of parameters. If indeterminate variations are desired, it may be mapped to any sort of modulator. Among the parameters, three stand out as novel developments: freely definable step sizes, and two syncopation parameters which I have named Irrationality and Eccentricity.

2.1 Calculating Subdivision Size

The step size of LR.step is entered to a text field in the Max note value format, an integer with a suffix of either 'n', 'nd', or 'nt' to indicate regular, dotted or triplet note length. LR.step circumnavigates a limitation of the note value system that the subdivision be an integer power of 2. With the LR.step sequencer, any integer up to 128 may be given followed by any of the suffixes. This is accomplished by reference to another timing system common to digital audio workstation applications: 'ticks'. Ticks are consistently 480 per quarter note. They remain constant relative to rhythmic values, not relative to the pulse of the time signature denominator, which makes it possible to compare the ticks per an entire measure in any given meter. If a quarter note is a subdivision of a whole note into 4 and a half note is a subdivision of a whole note into 2, a fifth note is simply a division of the whole note into 5.

To find subdivisions other than the ones available with the Max timing objects, the number of ticks in a whole note (consistently 1920 or 4 * 480 ticks) may be divided by the number of the subdivision symbol to find duration in ticks, for example 1920/4 = 480. 1920/37=59.$\overline{891}$. In the case of 'nt' or 'nd' subdivisions, this value is then multiplied by 0.$\overline{6}$ or by 1.5 respectively. The length of the entire sequence is calculated as the number of measures to sequence, multiplied by the time signature numerator, multiplied by the tick length of the time signature denominator as described above. To calculate the number of steps in the sequence, we divide the total sequence length by the individual step length.

This, combined with the idea of simply restarting the sequence at the end of its measures regardless of having fully completed the final step, opens the possibility of accommodating many new step sizes within a whole number of measures. For example – one 4/4 measure with a dotted eighth note steps size truncates the final pulse after a 16th note's duration. The fractional number of steps then scales the sequence driver (a signal ramp from a hostphasor~ object) by the total decimal number of steps to index the onsets of the steps accordingly. To provide visual feedback, the user interface rounds up the step count and scales the width of a multislider object behind a panel object to visually truncate the last step.

Figure 2. A sequence of dotted 8th notes in 4/4

2.2 The Eccentricity Parameter

Once the number of steps in the sequence has been calculated, it is further transformed by the Eccentricity parameter before being solved for the table of indispensability priorities. Eccentricity is a decimal number between 0 and 1, representing a range from 100 percent to 200 percent of the number of steps in the sequence. This scaled number of steps is rounded up to the next integer and solved for the lowest prime factors (as per Barlow's definition of meter stratification) for sequences up to 10000 steps. The list of prime factors is what is actually input to Hadju's Dispenser external, which then outputs the list of metrical weights. An indispensability set for a sequence some percentage longer than the actual playing sequence is calculated and then truncated to the sequence length.

The eccentricity parameter provides a simple means of generating syncopation; for example, taking a straight 4 on the floor pattern of 16th notes and transforming it into a dotted quarter, dotted-quarter, quarter note pattern at 150% of the 4/4 sequence. This method also achieves more oblique syncopations at longer sequence lengths and more slightly shifted eccentricity percentages, e.g. a sequence with 32 steps and an eccentricity value of 0.2 is actually solving the indispensability for a sequence of 39 steps and truncating that list to 32 steps.

Figure 3. Two sequences of different eccentricities

2.3 The Density Parameter

The density parameter is a float between 0. and 1. that represents the percent of steps in the sequence to actually trigger. A change in the density parameter clears the output sequence and then indexes through the steps of the latest calculated indispensability set (by their priority, not by position) for the fraction of steps it is set to, updating the output sequence for each step that it reaches. In a sequence with 16 steps, a density of 0.25 results in only the four most important steps being present.

Figure 4. A sequence at two densities

2.4 The Irrationality Parameter

Termed 'Irrationality', this processing stage changes the order at which steps appear as the density increases. Irrationality does not change the topology of the step priorities but rather which combinations of hits are present at densities less than 100%. To accomplish this, the index to add as density increases is multiplied by a number irrational to and less than the total number of, wrapped around the sequence length by a modulus operator. It is particularly salient that the scaling of the density index be irrational to the sequence length because this produces complete sets of steps at full density, much like how a cycle of 4ths or 5ths forms a complete set of semitones in 12-tone-equal-tempermant harmony. An evenly divisible multiplier would output the same position multiple times at full density.

Figure 5. Two sequences of different irrationalities

2.5 Phase Shift and Reverse

A final phase shifting stage accommodates polyphonic drum patterns. For example, a snare backbeat in 4/4 (or two downbeats a half note apart) is most efficiently calculated with values of zero for both eccentricity and irrationality, but at the density producing two events within a measure of 4/4, this results in hits on beats 1 and 3 rather than 2 and 4. Shifting the phase by 25% re-locates these steps to the backbeat.

The phase parameter is a floating point number between -1. and 1., representing twice the number of total steps in the sequence. Negative values for this parameter entirely reverse the list in addition to shifting the phase backwards, allowing for both a 0-100% shift forwards and a reversed sequence 0-100% phase shift.

Figure 6. A sequence with two different phase shifts

3. CONCLUSION

The LR.step sequencer presents a vast amount of capability of use to the computer music community. The Eccentricity and Irrationality processes are simple but functional extensions of Barlow's indispensability algorithm that generate many new patterns still ground in metricality. Furthermore, the approach to arbitrary step-sizes incorporating the Max note value format is a novel technique which may be implemented in other sequencing applications.

In the taxonomy of sequencer interfaces proposed by Duignan, Noble, and Biddle, Honeybadger would be classified as a flexible special purpose sequencer which provides a high degree of abstracted control and delayed linearization in a data flow system, meaning that this sequencer is designed specifically for electronic drum performance, and generates entire sequences in real-time with adjustments of only a few parameters [8].

3.1 Future Work

As a ReWire enabled Max patch running externally to the host sequencing program, the timing accuracy of the sequencer could be improved. The combination of ReWire and MIDI recording in Live adds some latency and a slight amount of inconsistency. Further development of this sequencer and interface will involve porting the LR.step patch into Max for Live, and additionally, creating a standalone mobile music app.

In Max for Live, more direct integration of transport timing through the plugphasor~ object will hopefully achieve greater precision. Alternately, as a Max for Live device LR.step might write and delete midi notes directly into an Ableton clip if the rapid recalculation of sequences does not run into a bottleneck while procedurally editing the clip.

The creation of a mobile app will provide further advantages. The functionality of a smartphone provides a compact wireless form-factor, gestural and reconfigurable touch control. Furthermore, mobile app publishing platforms will make the distribution and utilization of the sequencer much more accessible to a wide audience.

4. REFERENCES

[1] S. Booth and R. Brown, "Autechre Patch," in CYCLING 74 FORUMS Online Discussion Forum, 28 May 2008. [Online]. Available WWW: https://cycling74.com/forums/topic/autechre-patch/.

[2] G. T. Toussaint, "The Euclidean Algorithm Generates Traditional Musical Rhythms," in *Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science*, Banff, Alberta, 2005, pp. 47-56.

[3] H. Järveläinen, "Algorithmic Musical Composition," in *Seminar on Content Creation*, Helsinki, Finland, 2000, Tik: 111.080.

[4] C. Barlow, "Two Essays on Theory," *Computer Music Journal*, vol. 11, no. 1, pp. 44-60, 1987.

[5] G. Hajdu, "Automatic Composition and Notation in Network Music Environments," in *Proceedings of Systems, Mand and Cybernetics conference*, 2006, Available: http://www.smc-conference.org/smc06/papers/15-Hajdu.pdf [Accessed Feb 2016].

[6] G. Sioros and C. Guedes, "Generation and Control of Automatic Rhythmic Performances in Max MSP," in *Proceedings of the Simposio de Informàtica*, 2011.

[7] A. Eldridge, "Generative Sound Art as Poeitic Poetry for an Information Society," in *Proceedings of the International Computer Music Conference*, Ljubljana, Slo, 2012, pp. 16-21.

[8] M. Duignan, J. Noble and R. Biddle, "A Taxonomy of Sequencer User-Interfaces," in *Proceedings of the International Computer Music Conference*, Barcelona, 2005, pp. 725-728.

EVALUATION OF A SKETCHING INTERFACE TO CONTROL A CONCATENATIVE SYNTHESISER

Augoustinos Tsiros, Grégory Leplâtre
Centre for Interaction Design
Edinburgh Napier University
10 Colinton Road, EH10 5DT
a.tsiros@napier.ac.uk,g.leplatre@napier.ac.uk

ABSTRACT

This paper presents the evaluation of Morpheme a sketching interface for the control of sound synthesis. We explain the task that was designed in order to assess the effectiveness of the interface, detect usability issues and gather participants' responses regarding cognitive, experiential and expressive aspects of the interaction. The evaluation comprises a design task, where participants were asked to design two soundscapes using the Morpheme interface for two video footages. Responses were gathered using a series of Likert type and open-ended questions. The analysis of the data gathered revealed a number of usability issues, however the performance of Morpheme was satisfactory and participants recognised the creative potential of the interface and the synthesis methods for sound design applications.

1. INTRODUCTION

Morpheme[1] is a sketching interface for visual control of concatenative sound synthesis (see [1]) for creative applications. In recent years a number of user interfaces have been developed for interaction with concatenative synthesis [2]–[5]. Furthermore, although sketching has been widely explored as a medium with interaction with sound synthesis and musical composition (see [6]–[10]) there have been very few attempts to evaluate the usability of such interfaces. Additionally, Morpheme is in our knowledge the first attempt ever made to use sketching as a model of interaction for concatenative synthesis.

The way concatenative synthesis works is different to that of most conventional sound synthesis methods. Unlike other synthesis methods were the sound is represented by low-level signal processing parameters which can be controlled in a continuous manner, in concatenative synthesis, sounds are represented using sound descriptors related to perceptual/ musical parameters, and sounds are synthesised by retrieving and combining audio segmented from a database. Although this is a very interesting way of synthesising audio, it can lead to unexpected results, particularly for users that are not familiar with this type of sound synthesis. For example, while in other synthesis methods, increasing the amplitude parameter results in changes only to the parameter that was controlled, in the context of concatenative synthesis requesting a sound of greater or smaller amplitude may result in selecting different audio units that have very different timbre characteristics. These sudden/discreet changes could potentially confuse practitioners that are not familiar with this synthesis method.

The aims of the study presented in this paper are the following:

1) Evaluate Morpheme's graphical user interface: detect usability issues and identify desired functional requirements.
2) Evaluate the mapping between the visual features of the sketches and the control parameters of the concatenative synthesiser
3) Assess whether the audio used in the corpus affects the perceived level of control of the interface, the appreciation of the system and the mapping.

[1] Download Morpheme: https://inplayground.wordpress.com/software/

2. MORPHEME

2.1 Graphical User Interface

Figure 1. shows a screenshot of *Morpheme's* main graphical user interface. We could distinguish between four main interface components in the second version of *Morpheme's* interface, the canvas, the timeline, the playback controls, the brush controls and the video display.

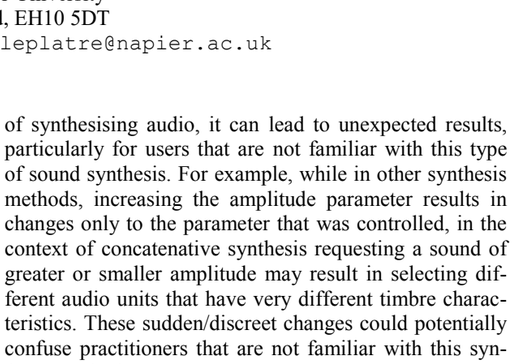


Figure 1. Morphemes' main graphical user interface.

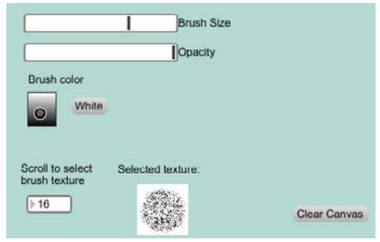
The playback controls provide a number of function (see **Figure 2**) including:

- **Play**: starts the analysis of the sketch which results in the data used to query the database and drive the sound synthesis engine.
- **Loop:** repeats the entire length of timeline when the cursor reaches at the end of the timeline.
- **Scrub:** functions freezes the cursor in a given location of the timeline. Dragging the cursor of the timeline can move the analysis window through the sketch to a desired position.
- **Speed:** allows the user to determine the speed (in milliseconds). The speed controls the rate at which the analysis window moves from left to right though the timeline.

Figure 2. Screenshot of the user interface playback controls.

Brush Controls provide a number of function (see **Figure 3**) including:

- **Brush size:** size of the brush
- **Opacity:** opacity of the textured brush.
- **Brush color:** color of the textured brush.
- **White:** control can be used as an eraser.
- **Brush selection:** by clicking and scrolling on the number box users can select from 41 different textured brushes.
- **Clear Canvas:** erases the sketch from the canvas.



- **Figure 3**. Screenshot of graphical interface for the control of the brush parameters.

2.2 System Architecture

Figure 4 illustrates the architecture of Morpheme. During playback windowed analysis is performed on the greyscale version of the sketch. A window scans the sketch from left to right one pixel at every clock cycle, the rate of which is determined by the user. Only the areas of the canvas that are within the boundaries of the window area are subjected to statistical analysis. The window dimensions are determined by Window width by window height. The window width can be determined by the user, however the default size of the analysis window is 9 pixel wide by 240 pixel height. The analysis of the canvas' data matrix results in a four dimensional feature vectors that describes the visual attributes of the sketch and which is used as the target for querying audio-units from the CataRT's database.

2.1.1 Mapping Visual to Audio Features for Selection and Processing of Audio Units

In the current implementation of *Morpheme,* we can distinguish between two mapping layers. The first layer consists of a mapping between visual and auditory descriptors for the selection of audio units, see **Table 1**. The second layer consists of a mapping that associates the distances between audio and visual descriptors to the synthesis parameters, see **Table 2**.

Visual Features	Audio Features
Texture compactness	Spectral flatness
Vertical position	Pitch
Texture entropy	Periodicity
Size	Loudness
Horizontal length	Duration

Table 1. Associations between audio and visual descriptors.

Audio features	Synthesis parameters
Spectral flatness	Transposition randomness
Periodicity	Grain size and amplitude randomness
Pitch	Transposition
Loudness	Amplitude

Table 2. Mapping the distances between audio and visual feature vectors to synthesis parameters.

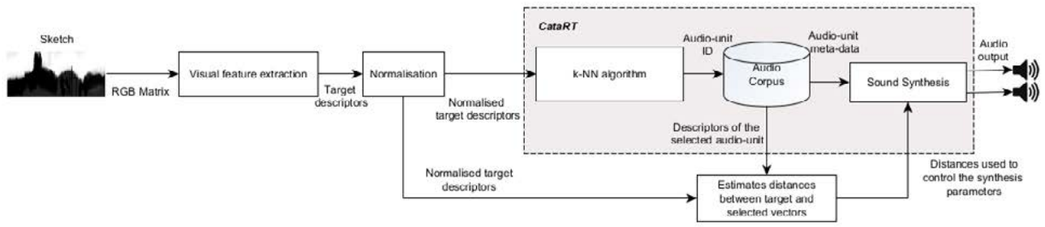


Figure 4. An overview of the architecture of Morpheme

3. MORPHEME EVALUATION

3.1 Participants

One group was recruited that consisted of eleven musician/sound practitioner volunteers. All of the participants played a musical instrument and the self-reported level of expertise was five intermediate and six advanced. Seven ofthe participants had received formal music theory training at least for six months. All of the participants reported using analogue and digital equipment for sound synthesis, signal processing and sequencing. Four participants self-reported a level of expertise regarding the use of digital and analogue equipment as intermediate and seven reported advanced skills. None of the participants in this study reported having hearing or visual impairments. All participants had first participated in the experiments described in the chapters five and six prior to taking part in the present one. All participants were male while the age group ranged from 18 to 64.

3.2 Apparatus

The experiments took place in the Auralization room at the Merchiston Campus of Edinburgh Napier University. Participants used Beyer Dynamics DT 770 Pro monitoring headphones with 20db noise attenuation to listen to the audio stimuli. An HP ENVY dv7 laptop with 17.3 inch screen was used. For sketching on Morpheme's digital canvas a bamboo tablet was used. However participants were allowed to use a computer mouse if they preferred. SurveyGismo was used to record the participants' responses after the sound design task was completed.

3.3 Procedures

In this study participants were asked to design two soundscapes using the *Morpheme* interface for two video footages. Subject responses were collected independently. In each session a single participant completed the following tasks. Participants were given a brief description of the task followed by a short demonstration of Morpheme's graphical user. After a short training session were participants were shown how to use the graphical user interface of Morpheme in order to synthesize sounds, participants were instructed to proceed with the tasks. There were two eight minutes sessions (one for each video footage) during which participants were free to produce a soundscape that best suited the video using Morpheme. At the end of the sessions, participants were asked to complete a questionnaire. The questionnaire consisted of 15 Likert type (i.e. 1= strongly disagree, 5 = strongly agree) an open-ended questions. The questions aimed at assessing experiential, cognitive and expressive aspects of the interaction as well as to detect usability issues and gather ideas regarding usability improvements of the interface.

3.4 Materials

3.4.1 Video footages

Two videos have been selected for this task. The first video footage has been captured in Bermuda during the recent hurricane Igor, see **Figure 5** top row. The duration of the hurricane video is one minute. The camera shots included in the video have been captured from several locations during the hurricane. The second footage is a 3d animated scene that last for 4 seconds which represents a simulation of two porcelain objects been shattered on a tilled floor, see **Figure 5** bottom row. Both video footage require a relatively high precision in the way the sound is synced to the video sequence. However the second video sequence is slightly more challenging in this respect in comparison to the hurricane scene.

Figure 5. Four screenshots from the two video footage used in the study.

3.4.2 Audio Corpus

The audio corpus that participants had to use to synthesize the sound effects for the shattering scene consists of four audio recordings of glass shattering events. The corpus that is used to synthesize the soundscape for the hurricane scene consists of four audio recordings of windy acoustic environments. All eight audio files have been segmented to audio-units with durations of 242 milliseconds. The selection of the audio files used to prepare the two corpus was predominately determined by the theme of the video footage. However these two videos were selected to allow testing the mapping in two very different auditory contexts. For example the shattering scene requires a corpus that consists of sounds that are relatively dissonant, non-periodic, and abrupt such as impact/percussive sounds. The second hurricane scene requires a corpus that contain moderately harmonic, slightly periodic and continuous sounds.

4. RESULTS

The first question aimed at assessing participant satisfaction of the sounds created using Morpheme, see **Table 3**. The participants' average response shows that they were neutral regarding this question. Participants' responses show that there was a strong correlation between the user input (i.e. sketch) and the outcome sound, and that it was easy to understand the mapping. Although the degree of correlation was not as strong at all times. Participants' responses indicate that Morpheme's sketching interface helped them articulate their sound design ideas in visual terms, and that they felt they had control over the sound synthesis parameters. However the responses also indicate that more precise control of the audio parameters would be desired. Participants felt equally in control using either corpus (i.e. wind and impacts) while there was indication that there was a stronger preference in working with the impacts corpus. Finally, participants agreed that Morpheme offers an interesting model for interaction with sound synthesis parameters and that it would be a useful addition to the sound synthesis tools they already use. An analysis of the data gathered by the open ended questions was performed manually. Every time a new theme was encountered in the answers, it was used to form a new *category*. Then the frequency of these categories was recorded to identify which the most prominent issues and desired technical features. The usability improvements identified are summarized in **Table 4**

5. DISCUSSION

Based on the results presented above, it can be concluded that overall Morpheme achieves a satisfactory level of performance. The subjective level of control of the sound parameters through sketching, and the participants' level of satisfaction with the sounds they designed was average. These results might be attributed to three factors. The first factor is the user's unfamiliarity with sketching as a model of interaction with sound synthesis parameters. The second factor might be their unfamiliarity with the way concatenative synthesis works. This view is further supported from the average responses (M=3 SD=1) to the question 'I felt confused in several occasions about how my drawing affected the audio output'. This is also reflected in some of the user comments, for example:

"Unpredictable results at times",

"It wasn't always easy to be precise",

"It was complicated at times to identify the correlation between the pitch and the type of sounds played".

As it was mentioned earlier in Section 3.2 the information that was provided to the participants prior to the experimental task was mainly about how to use the interface. Minimal information was provided about the synthesis method. This decision was made primarily to avoid the development of positive biases towards the system due to enthusiasm about the way the system synthesises sound. The third factor might be related to the usability issues identified.

Overall, the perceived correlation between the visual and sound features were satisfactory. Participants' responses showed that Morpheme is easy to use, offers an interesting approach to interacting with sound synthesis and supported that the interface helped them think about sound in visual terms. Furthermore, the majority of participants thought that Morpheme would be a useful

Suggested user interface improvements	N
Image processing tools for refinement of the sketch	1
Timestamps navigation of the timeline	2
Edit the position of graphics based on timestamps	1
Larger canvas	8
Canvas zoom-in function	6
Temporal looping function based on user defined loop points	1
Undo function	1
Latency between graphics and audio timeline	5
Non-linear sketch exploration	1
Enable layering of multiple sounds/sketches and ability to shift between layers	1

N=Number of participants

Table 4. Participants' answers to the question: What changes to the User Interface would you suggest to improve it?

	Questions	Mean	STD
1	I am satisfied with the sound I designed using this mapping.	3	0.85
2	I felt there was a strong correlation between the sketch and the sound that was synthesised by the system.	4.18	0.38
3	I felt I understood how attributes of the sketch were associated to attributes of the sound.	4.54	0.65
4	I felt I could articulate my creative intentions using this mapping.	3.9	0.51
5	I felt I had control over the synthesis parameters while using the system.	4.18	0.57
6	I am satisfied with the level and precision of the control I had over the audio parameters while using the system.	3	0.85
7	I felt confused in several occasions about how my drawing affected the audio output.	3	1.04
8	Overall, I am satisfied with Morpheme's Graphical User Interface.	4	0.42
9	I believe that Morpheme offers an interesting approach to interacting with sound synthesis.	4.81	0.38
10	I believe that Morpheme would be a useful addition to the audio tools I currently use.	4.45	0.65
11	I felt Morpheme helped me think about sound in visual terms.	4.27	0.86
12	I felt equally in control while using the two sound corpora.	3.54	0.65
13	I felt frustrated about certain aspects of the interface/interaction.	2.9	0.79
15	I felt that Morpheme was complicated and difficult to use.	1.9	0.5

Table 3. Statistics of the Likert type questions that were answered by participants (1= strongly disagree, 5 = strongly agree).

addition to the audio tools they currently use. Participants responses were not conclusive as to whether the corpora that was used affected their perceived level of control over the system as participants response was (M= 3.5 SD=0.6), while seven out of eleven participants seem to prefer working with the impacts corpus, three preferred the wind corpus and one neither. One of the differences between the impacts and the wind corpora is that the former is much larger. Based on the findings from the evaluation it appears that a larger corpus can result in both positive and a negative effects. Some of the negative effects became evident from some of the participants comments discussed above such as more unpredictable results, because the probabilities of getting a sequence of audio-units with very distinct timbre is higher when there is a large nonhomogeneous corpus (e.g. impacts corpus used for the evaluation) than when a small and homogeneous corpus (such as the wind corpus) is used. Furthermore, it is worth noting that participants' were moderately satisfied with the sounds which they designed using the system (M=3 SD=0.8).

Many usability issues were also revealed, mainly related to the lack of standard controls found in other image processing applications (e.g. photoshop) such as zooming in and out, resize canvas and undo function. Further, participants also pointed out the lack of other functions that tend to be standard functionality in time-based media production applications such as setting loop and cue points on the timeline, having a precise transport panel and a sequencer were sounds can be layered. Moreover, several participants complained about latency between the timeline and the output sound. Latency depends on two factors: the size of the audio corpus (i.e. how many audio units are stored in the corpus) and how many comparisons the algorithm has to perform until it finds the audio-unit that its features best match the target. Another factor that might cause the perception of latency is that in the present version of morpheme, the current position of the analysis window is indicated by a slider that does not reflect well the actual position of the window, see top image in **Figure 6**. The problem is that the window is 9 pixels wide while the current cursor used to represent the position of the analysis window suggest that the window is smaller. A better solution would be to use a cursor as shown in **Figure 6** bottom.

Figure 6. The top figure shows the current visual feedback for the representation of the position of the analysis window. The bottom figure shows a more precise visual feedback.

6. CONCLUSIONS

The evaluation of Morpheme showed that the performance of Morpheme was satisfactory and participants seemed to recognise the creative potential of the tool. From the analysis of the results, we could distinguish between two types of issues. The first type were issues related to the user interface. Most of the usability and functionality features that the participants noted could be relatively easily addressed with the implementation of standard controls found in other time-based applications, or in more advanced drawing packages. The second type were issues related to the type of sound synthesis used by the application (i.e. target based automatic selection synthesis using low and high level descriptions). Some of issues involved the unexpected transition between audio-units that sounded very different, which gave participants the impression of lack of control. In order to create sounds that are plausible variations of the original audio used in the corpus a degree of awareness not only of the micro but also of the meso and the macro levels of the sound is required. The issues identified through this evaluation will form the basis for future development of the *Moprheme* interface.

Acknowledgments
The authors would like to acknowledge the IRCAM's IMTR research centre for sharing the *CataRT* system.

7. REFERENCES

[1] S. B. Diemo Schwarz , Grégory Beller , Bruno Verbrugghe, "Real-Time Corpus-Based Concatenative Synthesis with CataRT," in *Digital Audio FX*, 2006, pp. 279–282.

[2] D. Schwarz and B. Hackbarth, "Navigating variation: composing for audio mosaicing," in *International Computer Music Conference*, 2012.

[3] M. Savary, D. Schwarz, and D. Pellerin, "Dirty tangible interfaces: expressive control of computers with true grit," *CHI'13 Ext.* , pp. 2991–2994, 2013.

[4] J. Comajuncosas, "Nuvolet: 3d gesture-driven collaborative audio mosaicing," *New Interfaces Music. Expr.*, no. June, pp. 252–255, 2011.

[5] E. Tomás and M. Kaltenbrunner, "Tangible Scores: Shaping the Inherent Instrument Score," *Proc. Int. Conf. New Interfaces Music. Expr.*, pp. 609–614, 2014.

[6] G. Levin, "The table is the score: An augmented-reality interface for real-time, tangible, spectrographic performance," in *proc. of DAFX - Digital Audio Effects*, 2006, pp. 151–154.

[7] M. Farbood, H. Kaufman, H. Line, and K. Jennings, "Composing with Hyperscore: An Intuitive Interface for Visualizing Musical Structure," in *International Computer Music Conference*, 2007, pp. 111–117.

[8] P. Nelson, "The UPIC system as an instrument of learning," *Organised Sound*, vol. 2, no. 01, 1997.

[9] J. Garcia, T. Tsandilas, C. Agon, and W. E. Mackay, "InkSplorer : Exploring Musical Ideas on Paper and Computer," *Proc. Int. Conf. New Interfaces Music. Expr.*, pp. 361–366, 2011.

[10] J. Thiebaut, "Sketching music: representation and composition," Queen Mary London, 2010.

Recorderology - development of a web based instrumentation tool concerning recorder instruments

Ulrike Mayer-Spohn
Elektronisches Studio Basel - FHNW Basel
Leonhardsgraben 52, 4051 Basel
contact@ulrikems.info

Keitaro Takahashi
Elektronisches Studio Basel - FHNW Basel
Leonhardsgraben 52, 4051 Basel
neoterize@mac.com

ABSTRACT

In this paper, we describe our instrumentation research for recorder instruments and its documentation method developed to operate as a web application. Our aim is to propose an application that enhances the knowledge and experiences of musicians, especially composers, about the recorder family and to encourage their creative activities. Furthermore we suggest proper notations, which enable composers to illustrate their musical ideas precisely and increases the efficiency of communication between musicians and composers. In our research, we analyze the correlations between the mechanisms and the actual results of sound production by means of four primary components (instrument model - air - mouth - fingers). This project is carried out through an interaction between artistic research involving collaborations with composers and musicians and scientific research including audio analysis, e-learning and data-mining. In our web application, we employ a large audio database to describe the mechanism of the playing techniques of recorders along with a graphic user interface with the aim of simplifying the navigation.

1. INTRODUCTION

During the last twenty years, the way in which people use computers has changed immensely. The use of computer and web based environments is integrated into the daily life for diverse purposes, such as communication, learning and leisure. Nowadays, the educational systems include the use of computers and internet into the teaching and learning environment as a means of extending or supplementing the face-to-face instruction.

Especially in the case of music education/training, diverse musical researches report especially concerning instrumentation have been distributed or documented using internet technology. The technology is able to distribute the documentation resources by employing varied interfaces to present the different data. Many of these web implementations are applied as an extension of a book or CD media and therefore, these web-sites have relatively simple structures.

It is easily predictable that sound examples possibly give us a profound knowledge and precise imagination of the timbre (sound color or sound quality) which is not sufficiently covered by the standard western notation system. However, when the instrumentation research involves more complicated issues and seeks to clarify questions of sound phenomena and their mechanisms, more advanced audible demonstrations are requested by musicians, both instrumental players and composers for artistic purpose. Furthermore, we assume that the design of an interface combining literal and audible documentation is significant and it could be more effective when it contains a familiar description system for the musicians, for example notation.

Our instrumentation research concerning recorder instruments, known as Recorderology [1] , aims to provide versatile musical experiences without meetings or rehearsals with musicians depending on the users demands. Our goal is to interpolate self-study into the exchange with other musicians and to expand creative possibilities by optimizing the time and energy used for instrumentation study.

In our instrumental research, we target the mechanism and timbres of various playing techniques. The recorder family consists of significantly more different sizes and models compared to other families of instruments [2] . Generally, this factor strongly increases the complexity of the correlations between their sound productions, playing methods, notation and composition. Therefore, it is important to discuss an efficient way to organize the diversity and complexity of the recorder family in a clearly structured interface combining music notation with sound samples.

In a further step, we address the development of a web documentation method by using an interactive user interface, a web Application with Web Audio API, which enables us to build an advanced signal processing program and an interactive audio sampler in hypertext documents without any plug-in. In our web application, we employed a large audio database and interactive data retrieval system in order to describe the details of our research results.

[1] The research project Recorderology is the second step of the project Recorder Map http://www.recordermap.com

[2] The recorder family consists of diverse sizes and types of instruments. Each single recorder generally produces a significantly different timbre due to the different inner bore and voicing (construction of the wind way and labium). The Medieval, Renaissance, and Baroque eras had their unique types of recorders and in each epoch they could come in up to nine different sizes.

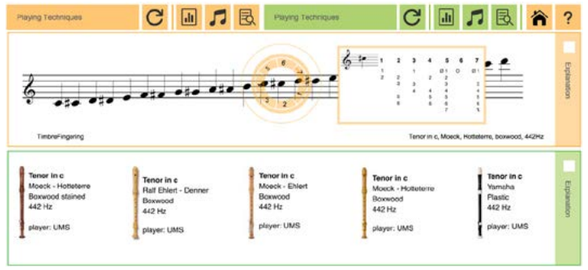


Figure 1. This figure illustrates the layout of the web-application Recorderology on Google Chrome browser

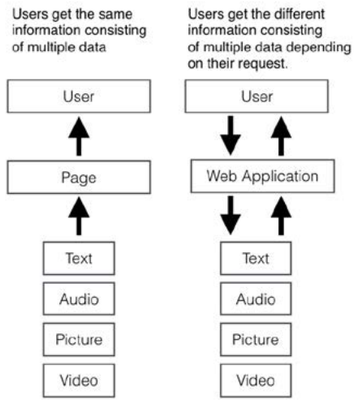


Figure 2. This figure represents the differences between the functionality of the mentioned web-sites and our web-application

2. OVERVIEW OF THE WEB DOCUMENTATION METHOD

We designed the web application Recorderology after an evaluation of its potential and effectiveness and inspecting existing web documents. We realize the importance of studying the expectable effect of a web documentation and of analyzing actual examples.

2.1 The effectiveness of e-learning

Some applications of internet technology in educational/ training modules are known as e-learning (electronic learning such as computer-based learning, online learning or distributed learning). Compared to Face to Face (FTF) instruction or paper based documentation, e-learning modules are more interactively adapted towards a particular goal depending on learner's demands.

Several research projects have already evaluated the effectiveness of e-learning. Tyechia (2014)[1] evaluated the comparative level of proficiency of learning between FTF learning and e-learning by comparing the scores of a candidate's paper test. His results suggest that both methods are equally effective or in some cases slightly positive for e-learning under his conditions.

Karachi and Ambekar (2015)[2] analyzed the effect of e-learning and attested a positive impact concerning the two facets Explanation and Interpretation, which are the two most fundamental of the six facets of understanding [3] .

Although the researchers noted that the effectiveness of e-learning is significantly influenced by the design of websites or applications, we can expect that e-learning can reach the same level of effectiveness as FTF learning, especially concerning fundamental understanding levels. The advantage of e-learning is its high responsiveness to different purposes and demands. As musicians tend to have unique demands in their creative work, the element of versatile and adaptable instruction is an important factor in musical studies.

2.2 Related examples

A comparable effect can be expected in the case of musical study concerning topics such as instrumentation, composition, organology and sound analysis. Here we survey examples of audio techniques used in web documentation of musical research as the implementation/presentation of audio data is a crucial factor in music study..

Flash player is one of the most often used plugins to present audio samples with or without a graphical user interface, for example in PRIME-project [4] , clarinet- multiphonics [5] , "The Techniques of Saxophone Playing"[3] [6] etc. Clarinet multiphonics employs Flash player to produce an interactive multiphonics chart. It presents fingering charts with their generated multiphonics specifying dynamic levels, pitch information, difficulty of performance, and sound examples. Users can select the specific multiphonics information by its fundamental pitch.

The HTML5 tags <audio>and <video>are also used as one of the simplest ways to present audio samples. Con Timbre [7] and the Academy page of Vienna Symphonic Library [8] provide two examples. Users need to load an individual audio file each time they listen to it.

Video is used to present the relationship between physical movements of a music performance and its sound result. In this case, video sharing services such as YouTube, Vimeo, etc. are often used to deliver a stable data flow and save on server storage. The videos are especially beneficial when they present extended or unusual playing techniques. One remarkable example is CelloMap [9] , by E. Fallowfield[4], which demonstrates the actions of a cello player and their sound results.

Figure 2 shows the functionality of most of the other mentioned cases compared to our case.

3. RECORDEROLOGY

Basing our work on fundamental artistic needs from collaborations with composers, we intend in our research project

[3] Grant Wiggins and Jay McTighe: the model of six facets of understanding consisting of Explanation, Interpretation, Application, Perspective, Empathy and Self-Knowledge

[4] http://www.primeresearch.ch

[5] http://www.clarinet-multiphonics.org

[6] https://www.baerenreiter.com/materialien/weiss_netti/saxophon/multiphonics.html

[7] http://www.contimbre.com

[8] https://vsl.co.at/de/Academy

[9] http://www.cellomap.com

Figure 3. Waveform on Google Chrome browser

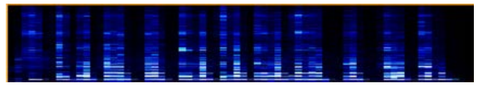


Figure 4. Spectrogram FFT analysis on Google Chrome browser

"Recorderology" to construct a database consisting of a large amount of material concerning various playing techniques using diverse recorder instruments (see Figure 1).

In our first step, we break down the playing methods into four main components [10] (model- air - mouth - fingers) and secondly analyze these components individually. Subsequently, we investigate the relationships between the components and the sound results and then determine how the diverse sounds produced are associated with the combinations of different components.

For example, the following list indicates the playing conditions for recording the samples using the playing technique "timbre fingering".

- fixed tuning pitch to equal temperament A4 = 442 Hz
- select desired instrument (referring to component model)
- blowing pressure adapted to achieve the exact pitch (referring to component air)
- articulation adapted to get the fingering sounding at the used blowing pressure (referring to component mouth)
- several preselected timbre fingerings (referring to component fingers)

Although the diversity of instruments indicates the huge artistic potential of the recorder, it is simultaneously an obstacle for comprehensive documentation. The web application of Recorderology provides a Graphical User Interface for studying the possible variations of several playing techniques. This web application presents the collected sample database arranged by the different instruments and components. Furthermore, the user can access score examples of the selected sound samples, playing instructions, notation and other descriptions. When the user selects one specific note, the program shows its possible variations of

[10] Wolfe, Almeida et al.[5] described that musical performance involves the interaction of the principal acoustical components in a wind instrument-player system:
source of air: the airflow is generally controlled by muscles of the torso and in some cases the glottis (referring to the component air). On the very short time-scale, the airflow is also controlled by the tongue, which can cease the flow by contact with the roof of the mouth (represented by the component mouth)
vibration element: the edge-tone produced at the labium (represented by the component model)
the downstream duct: the bore of the instrument (represented by the combination of the components model and fingers)

Table 1. The process of our Web application

process	GUI	behaviour
1	Select a PlayingTechnique from the menu	Show available instruments
2	Select a instrument	draw notes, description and load all requested audio data from database
3	Select a note	show a dynamic or variation menu
4	Select a dynamic and a variation	play a specific audio data
5	Close or Select Another PlayingTechnique or Select another instrument	delete loaded audio data and go back to **2**
6	Open Analysis window	draw waveform of previously played audio data and show analysis results

the playing technique as if a loupe is magnifying the details of the timbre. The selection menu appears in a circle around the note, which increases the visual focus to the standard notation, and at the same time the user can navigate around it to access different options of its timbre.

In the main page, two players are installed in parallel on the same window, and the user is able to assign a set of samples of different playing techniques or instruments to each player in order to investigate the sensitive differences between them.

4. SAMPLER SYSTEM - WEB AUDIO API IMPLEMENTATION

We defined a labeling rule, which identifies each audio file and illustrates its information about pitch, instrument, dynamic and so on, in order to simplify the data retrieval from our database.
The labeling rule:
instrument number_tuning pitch reference (Hz)_pitch number (Midi)_playing technique number_variation number_ dynamics (p or f)

Following the labeling rule, the name of audio file and buffer object in Javascript are defined as shown in the following examples:

An example of audio file name
15_442_77_3_1_p

An example of buffer object in Javascript
Audio_15_442_77_3_1_p

Each audio file is stored on the server in mp3 format in order to reduce the cost of internet communication and server storage. The files are then requested by using XMLHttpRequest and are decoded to raw data by AudioContext.decodeAudioData(). A bunch of audio files of a particular playing technique for each instrument is loaded to each corresponding buffer object at once [11] (in process 2,

[11] The multiple sound file loading/converting system was based on

Table 1) and are stored until they are discarded (in process 5). Therefore, the web application does not need to load an individual file for each time of playing and this enables users to access the same group of audio files instantly and to easily inspect the detailed distinctions or variations of different audio files. The program illustrates the waveform (Figure 3) of the loaded audio data and its spectrogram (Figure 4).

5. PROVISORY TEST RESULTS

Although the web application Recorderology is still in the development process, it is being tested by several composers over the world in a number of projects. The composers use it for investigation of recorder instruments and to prepare sketches of their new compositions before the first meeting with the musicians.

The first feedbacks show, that the presentation of the audio samples in context with the notation improved significantly the understanding of the various timbres of the instruments.

The figures 5-7 are score examples extracted from actual compositions written during this project. Variable playing techniques introduced in Recorderology are applied and developed within these compositions. Their notation is also based on our suggestions.

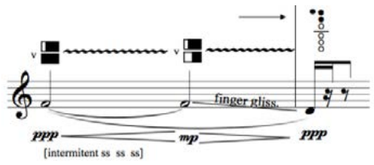


Figure 5. Luis Codera Puzo: Oscillation ou interstice (2013)

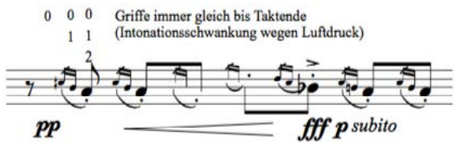


Figure 6. Christophe Schiess: Once estaciones (2014)

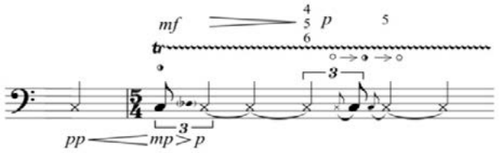


Figure 7. Keitaro Takahashi: surge (2014)

6. CONCLUSIONS

Our web application offers an interface covering individual playing techniques broken down into the components and their possible variations. We expect this application to enable users to develop their understanding and increase their experience of the instruments in a way that will eventually stimulate them with new artistic ideas.

So far, we have attempted to describe the differing playing techniques based on the combination of the components. However, our sampler system does not cover all of the small variations and users have to investigate these by themselves.

In a further development of this application, we intend to implement an automatic data retrieval module, which represents related playing techniques, variations, and score examples from various contemporary compositions, based on the criteria of audio analysis and audio categorization. This step is intended to interpolate detailed sound variations modified by the different combinations of components in a fast and convenient manner.

The Recorderology has the potential to be applied to other instrument families such as strings, keyboards, brass instruments etc.

The current version of our web application is available from the link below:

http://recorderology.com

7. ACKNOWLEDGMENTS

We give thanks for support to the Forschung und Entwicklung Dept. of FHNW Basel and gratefully acknowledge the financial support of Maja Sacher Foundation for the RecorderMap project during 2013-2014.

8. REFERENCES

[1] Tyechia, V.P. ”An Evaluation of the Effectiveness of E-Learning, Mobile Learning, and Instructor-Led Training in Organizational Training and Development”, *The Journal of Human Resource and Adult Learning* Vol. 10, Num. 2, December, 2014

[2] Kamatchi, R. and Ambekar, K. ”Analysis of e-learning web application’s alignment with six facets of understanding”, *International Conference on Advances in Computer Engineering and Applications (ICACEA)* IMS Engineering College, Ghaziabad, India, 2015

[3] Weiss, M. and Netti G. ”The Techniques of Saxophone Playing”, *Bärenreiter-Verlag* Kassel, Germany, 2010

[4] Fallowfield, E. ”Actions and Sounds - An Introduction to Cello Map”, *dissonance 115*, p. 51-59, September, 2011

[5] Wolfe, J.and Almeida, A.and Chen, J.M. and George, D. and Hanna, N. and Smith, J. ”The Player-Wind Instrument Interaction”, *School of Physics, The University of New South Wales, Sydney, 2013*

[6] B. Smus ”Web Audio API”, *O’REILLY* p. 11 - 12, 2013

[7] Arkorful, V. and Abaidoo, N. ”The role of e-learning, the advantages and disadvantages of its adoption in Higher Education”, *International Journal of Education and Research* Vol. 2 No. 12 December, 2014

“Putting it All Together“ written by B.Smus [6]

Karlax Performance Techniques: It Feels Like…

D. Andrew Stewart
Digital Audio Arts
University of Lethbridge
andrew.stewart@uleth.ca

ABSTRACT

The invention of performance techniques for the karlax digital musical instrument is the main subject of this paper. New methods of playing the karlax are described, in addition to illustrating new music scoring procedures for instructing the music practitioner how to play the karlax. An argument is made for a choice of language that is more familiar to the music practitioner, especially with respect to describing a potential for musical expression. In order to exemplify an approach to wording, which may be easily understood by practitioners, performance techniques and notation for a new composition, entitled "Ritual", are rigorously described. For instance, techniques are explained in relation to: the required physical gestures, notational symbols, audible output, and technical details. Furthermore, the performance techniques are organised into three categories: initiating a sound, controlling volume, and modulating timbre. Emphasis is placed on describing bodily awareness, achieving a holistic mode of interaction, and listening in an effort to convey the emotional undertones embedded in the music. By following these instructions for performance with the karlax, the practitioner will play with feeling, and not merely press keys, push buttons, and turn potentiometers.

1. INTRODUCTION

In this article, I endeavour to illustrate how my solutions to karlax performance techniques and notation reflect the phrases, "It feels like…" and "Do it this way" (see **2.1**, below). That is to say, my approach is directed at the music practitioner and especially the performer who regards the musical score as a means of interpreting a composition's emotional undertones, which are embedded by the composer through a notation system for employing traditional and new abstract symbology (e.g., the musical note, dynamic indictors, articulation and tempo marks, etc.). Moreover, my approach is for the practitioner who understands that a notation system, which frames instrumental playing techniques (i.e., identifying possible gestures), also frames a composition's musical ideas, which are conveyed by the creation and modulation of complex sounds.

2. MUSICAL EXPRESSION

A real-world illustration of a digital musical instrument always helps to frame a discussion around musical expression. For instance, audiences would find it difficult to disregard the expressive intentions of the performer in my karlax solo, entitled *Toward a ritual* (2015)[4]. The first few minutes provide an example of an intimate, minimal soundscape and include a theatrical "anointing" of the public. An audience could not possibly disregard the performer's intention to communicate an emotion embedded in the music.

Regarding a performance as the communication of emotions is quite common. For instance, communicating emotions is implied when performers define expression in music with phrases such as "playing with feeling"[6]. In addition, musicians judge their own performances as optimal when they (the performers): (1) have a clear intention to communicate [usually an emotional message]; (2) are emotionally engaged with the music and; (3) believe the message has been received by the audience[9]. Yehudi Menuhin, the renown violinist, conveyed this idea very clearly: "Unless you think of what the music carries, you will not convey it to the audience"[8]. We can easily imagine the master teacher instructing young pupils to "play with feeling" with phrases such as "It feels like…" and "Do it this way" and not "This is how it works" or "Think of it this way"[11].

Herein, lies an important distinction between music practitioners and others who may lack performance experience or who may wish to qualify (and quantify) expressivity in music objectively – detached from the communication of emotions. The active practitioner speaks with phrases such as "It feels like…" and "Do it this way". The non-practitioner may identify and define expression in music with phrases such as "This is how it works" or "Think of it this way".

2.1 Do it this way or think of it this way

Presently, unpacking expression in music within the context of new paradigms for integrating technology and musical performance and especially, performance techniques for digital musical instruments, is complex[2]. Analyses of expression are divergent and traces of both perspectives, "Do it this way" and "Think of it this way", exist in the literature.

Within the new interfaces for musical expression (NIME) community, "Do it this way" may be represented by a thoughtfulness toward human gesture and the communication of emotions facilitated by gesture. For instance, instrumental gestures are defined by Cadoz as an

analog of articulatory gestures, transferring energy to an object and transmitting expressive content to the audience[1]. Ryan identifies tangible physical effort as a significant aspect in the perception of expression[13]. Yet other authors provide a more abstract discourse around embodiment and musical expression. In his 2006 manifesto, *Enchantment vs. Interaction*, Waisvisz describes the "body as [a] source of electrical and musical energy"[15].

Other NIME community members, who may represent "Think of it this way", commonly use the terms "expression" and "expressivity" as part of their standard discourse. In some cases, they appear to liken expression to technological methods and even to the technology itself. For example, Roven et al. consider gesture acquisition (i.e., software-based mapping strategies) between the interface and performance gestures as a central determinant of expressivity[12]. Iazzetta's examination of musical gestures also points to mapping strategies (e.g., one-to-one versus convergent) and the author theorises that the degree of expressivity may be inherently linked to the complexity of the strategies being used[5]. Furthermore, researchers in the NIME community have qualified the object-technology, itself, as intrinsically expressive[3,10,14,16].

It is important to point out that the language used to describe expressivity in the NIME field may have a substantial function in dealing with the dialectic of musical expression. That is to say, while some authors may attribute expression to an object-technology, I find it difficult to believe that the same authors would disregard the human performer's potential to imbue a performance with musical expression.

3. KARLAX

The karlax resembles a clarinet or soprano saxophone in size and geometry, although its control structures do no involve blowing air through the instrument.[1] Instead, the karlax wirelessly transmits data to a sound engine (i.e., computer software instrument) by manipulating 10 keys (with continuous range output), 8 velocity-sensitive pistons, 17 buttons and a combination mini-joystick and LCD character display, operated with the thumb of the left hand. The interior of the karlax contains both a 3-axis gyroscope and 3-axis accelerometer. In addition, the upper and lower half of the karlax can be twisted in opposite directions; that is to say, the upper and lower half can be rotated in opposite directions because the joint between the two halves of the instrument acts as a type of rotary potentiometer with a maximum rotation angle of 65°. Furthermore, at each angle boundary (i.e., 0° and 65°), the karlax offers an additional 12.5° of resistive twist space, providing a resistive force for the performer, who may have a sensation similar to bending or pulling a spring – albeit the movement is still a twisting/turning motion.

[1] Development on the karlax began in 2001. This digital musical instrument has been commercially available since approximately the mid-2000s and is manufactured by DA FACT, in Paris, France. http://www.dafact.com/

Figure 1. Karlax digital musical instrument.

3.1 Karlax techniques, mappings and notation

In 2015, I developed several new musical pieces for the karlax, showcasing new performance techniques and parameter mappings. My work was especially influenced by the music scoring – notation – concepts of Mays and Faber,[7] as well as by the ideas shared at the first-ever Karlax Workshop, which took place at CIRMMT, McGill University (Montreal, Canada), in May, 2015.[2] Importantly, based on my experiences at the workshop, I developed new karlax techniques, mappings, and notation that expanded on the work of other participants, as well as inventing my own unique approach to creating and modulating sound with the digital instrument.

4. RITUAL: THE SCORE

Ritual (2015) is my first fully-notated musical composition for the karlax. The notational style resembles traditional music, with the addition of custom-designed symbols for conveying both the sound result, which I refer to as "musical gestures", and the manoeuvring of the instrument, which I refer to as "performance gestures" or "playing techniques" – examining the definitions of gesture are beyond the scope of this article.

"It feels like…" and "Doing it this way" are intimated by a set of performance tips given on the first page of the score. These tips do not function to explicate karlax playing techniques. Rather, they are added to the score in order to encourage a specific attitude or disposition toward exploring musical and performance gestures. With these initial suggestions, I encourage the performer to play with feeling.

• Your interpretation will be unique. The flexibility inherent in karlax playing techniques, as well as the sounds produced by these techniques, will create a novel effect that will be characteristic of your interpretation.

• Strike a balance between a flexible approach to techniques/sounds, and producing the 'same' piece at each rehearsal and performance. In other words, strive to perform the essentials of the composition in the same way, each time. For example, strive for a similar balance of loudness and timbre in multitimbral sections of the composition – creating textures.

[2] Workshop description/schedule: http://www.cirmmt.org/activities/workshops/research/karlaxWorkshop2015/event

• Concentrate your attention on exploring the sound initiation and modulation techniques and especially studying how to create and modulate complex timbre morphologies. Give yourself the freedom to explore the rich range of timbre possibilities.

• Music notation does not definitively describe musical gestures and so, always consider your choices and possible options with respect to certain notational symbols. For instance, the karlax tablature grids are used minimally only as reference points – starting points at which a sound is initiated. Consequently, do not constrain yourself to maintaining the grid indications throughout the composition. The same approach should be considered for shaking (e.g., "Sustain sound by lightly shaking from end to end") and twisting (e.g., "Twist elbows out").

5. RITUAL: TECHNIQUES

This section describes the principal playing techniques required for performing *Ritual* for karlax. The techniques are described in relation to the following topics: the required physical gestures, notational symbols, information related to traditional forms of music notation, audible output, and any necessary technical details. The techniques are organised into three categories: (1) initiating a sound, including muting/silencing [**5.1-5.2**]; (2) volume control [**5.3**]; (3) timbre modulation [**5.4**]. Some techniques correspond to multiple categories. However, an effort has been made to identify each technique's primary function by placing it into a specific category. The accompanying graphics are extracted from the score.

5.1 Initiating a sound with keys

The karlax keys are used similarly to the keys of a piano keyboard in this composition. They are treated as discrete on and off signals. The left-hand keys are notated on the top four-line staff, while the right-hand keys are notated on the bottom staff. Furthermore, each "space" on the staff, including the space above the topmost line, is assigned to a key with the exception of the lowest space, which is assigned to the two pinky keys (see **5.1.1**). For example, in the accompanying graphic (**Figure 2**), the middle finger and the pinky of the left hand play a two-note chord, while the right hand plays several staccato notes with different fingerings.

Unlike traditional music notation, the position of note heads does not correspond to pitch (or frequency space); for example, a note head in the lowest space does not necessarily produce a low frequency. However, in combination with octave selection (not described in this document), the fingering system yields higher tones when fewer keys (or a single key) are depressed and lower tones when more keys are held down. A fingering chart, which illustrates a correspondence between fingerings and pitch, is provided in the controller patch (in Max).[3]

[3] *Max* graphical programming environment for music, audio, and media. https://cycling74.com/

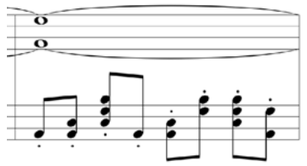


Figure 2. Initiating a sound with keys.

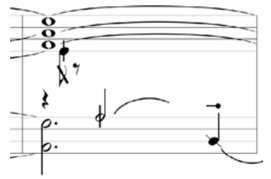


Figure 3. Initiating a sound with pistons.

Left-hand keys produce a flute-like pitch material in this composition. Right-hand keys produce a wide range of sounds from small bell-like (and bouncing) percussive tones to a pitched scraping or noisy blowing sound. Although the keys are used to produce discernible musical pitch, pitch frequency is approximate and microtonal.

5.1.1 The pinky keys

A note in the lowest space on the staff can be played by holding down either of the pinky keys. When to use the top (pinky 1) or bottom (pinky 2) key is indicated in the score. Each of the two pinky keys produces the same result. This permits some fingering flexibility and the possibility of alternate fingerings that may be useful in scenarios where the pinky finger is used to hold down both a key and piston at the same time. Moreover, because the two pinky keys produce the same result, a very fast repetition of the same sound/pitch may be created by alternating between the two keys – similarly to a traditional pitch "trill" gesture.

5.2 Initiating a sound with pistons

Holding down a piston produces no audible effect. Instead, sound is initiated through a combination of selecting a sound/note by holding down a specific piston and then, thrusting or gyrating (rotating) the karlax. The following sections (see **5.2.1-5.2.4**), describe this holistic combination-gesture.

Left-hand pistons are notated on the top four-line staff, while right-hand pistons are notated on the bottom staff. Furthermore, each "line" on the staff is assigned to a specific piston. For example, in **Figure 3**, firstly, a piston is depressed with the pinky of the left hand in the accompanying graphic. Next, a piston is held down with the right-hand index finger. After that, the right-hand ring finger is used for the final piston. The graphic also indicates that

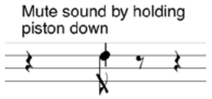


Figure 4. Piston thrust termination.

keys, which are illustrated by the note heads in staff "spaces", may be depressed simultaneously – a challenging fingering manoeuvre that produces multiple voices and must be practiced.

5.2.1 Piston: thrust onset

This technique requires a coordination of gestures: (1) holding down a single piston and (2) thrusting the karlax in the direction of the right hand – toward the end of the instrument that contains the on/off button. A thrust onset generates a realistic bell tone in this composition. Furthermore, a bell tone can be sustained indefinitely if "sustain mode" has been activated (sustain mode is not described in this document) and the performer continually shakes the instrument from end to end. Most importantly, the piston must be immediately released after initiating the sound in order to create a sustained bell tone. If not released, the holding down of a piston activates the "termination" of the note (see **5.2.2**).

A vertical line intersecting a note head is the symbol for thrusting (see **Figure 3**). An open-ended slur is attached to the note head in order to remind the performer to release the piston immediately. For onsets created by gyrating/rotating, read section **5.2.3**, below.

5.2.2 Piston: thrust termination

Similarly to the thrust onset, the thrust termination requires a coordination of gestures: (1) holding down a single piston and (2) thrusting the karlax in the direction of the right hand. In other words, the thrust termination produces a sound, as well as causing the sound to gradually decay (see Piston mute [**5.2.5**] for instructions on silencing a sound without triggering it). Hold down the piston for a slightly "extended" duration to terminate the bell tone. With practice, the correct extended duration will be perceived. Technically speaking, the required duration for depressing the piston is proportional to the strength of the thrust. A weak thrust causes the termination to occur more quickly and a strong thrust, more slowly – a range of between 300 and 1,500 milliseconds.

The notational symbol for a thrust termination gesture is a vertical line intersecting the note head. In addition, the termination gesture appears as an 8th note with a slash cutting through the flag of the 8th note stem (see **Figure 4**). Because this gesture produces bell tones, the thrust termination does not produce an instant silencing of the sound. Instead, the termination gesture produces a naturally decaying bell tone.

Figure 5. Four possible directions for initiating a sound *via* gyration: forward, backward, right, left.

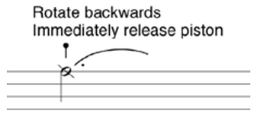


Figure 6. Piston gyrate onset.

5.2.3 Piston: gyrate onset

This technique requires a coordination of gestures: (1) holding down a single piston and (2) gyrating (rotating) the karlax – spinning the karlax like a baton around its central axis. Spinning can occur in four directions. The accompanying graphic of four dots, each with a short line extending away from the dot, illustrates the four possible directions: forward (up), backward (down), right, left (see **Figure 5**). From the perspective of the performer, this technique may be perceived as thrusting the left hand forward, backward, to the right, or to the left. A sustained bell tone is created by immediately releasing the piston after initiating the sound similarly to thrust onsets. If not released, the holding down of a piston activates the "termination" of the note (see **5.2.4**).

The notational symbol for a thrust termination gesture is a diagonal line intersecting the note head. In addition, the dot-line symbol (**Figure 5**) appears above a note head and indicates the direction of rotation. An open-ended slur is attached to the note head in order to remind the performer to release the piston immediately (see **Figure 6**).

In this composition, left-hand gyrate onsets generate realistic bell tones, while right-hand gyrate onsets generate imaginary bell tones – imaginary with respect to onset timbre and impact behaviour, as well as a complex decay structure that in and of itself, generates additional bell tones. Gyrate onset has the potential to sustain a sound indefinitely in the exact manner as "thrust onset", described above. However, due to the nature of the different bell tones (i.e., realistic versus imaginary), the result of a sustained sound is unique for each hand. With respect to the left-hand onset (realistic bell), the behaviour of the sustained sound is exactly the same as the tone generated by a thrust onset. With respect to the right-hand onset (imaginary bell), the sustained aspect of the sound takes the form additional bell tones – similarly to a conglomeration of bells that are continually struck.

5.2.4 Piston: gyrate termination

Similarly to the gyrate onset, the gyrate termination requires a coordination of gestures: (1) holding down a single piston and (2) gyrating (rotating) the karlax. The gyrate termination produces a sound, as well as causing the sound

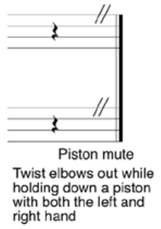


Figure 7. Piston mute.

to decay (see Piston mute [**5.2.5**] for instructions on silencing a sound without triggering it). To terminate the bell tone, hold down the piston for a slightly "extended" duration similarly to "thrust termination", above. The required duration for depressing the piston is proportional to the strength of the rotation.

The notational symbol for a gyrate termination gesture is a diagonal line intersecting the note head coupled with the symbol for rotation direction. In addition, the termination gesture appears as an 8th note with a slash cutting through the flag of the 8th note stem.

5.2.5 Piston mute

This technique silences sounds initiated by both thrust and gyrate onsets. Muting a sound requires a combination of gestures: (1) holding down piston(s) and (2) twisting the karlax into its region of maximum torque. Muting bell tones initiated by the left hand requires depressing any piston(s) with the left hand and twisting the upper and lower half of the karlax in opposite directions – twisting in a manner by which the elbows naturally move outward. Moreover, nearly the maximum angle of twist must be applied. Muting bell tones initiated with the right hand requires holding down any piston(s) with the right hand and applying the same twisting technique.

A double slash across the top staff line indicates a piston mute. A double slash across the upper staff denotes a left-hand piston mute and across the lower staff, a right-hand piston mute (see **Figure 7**). With respect to realistic bell tones, this technique causes tones to decay gradually and naturally. With respect to imaginary bell tones, muting is immediate.

5.3 Volume control

5.3.1 Twisting-tilting

The volume (loudness) of sounds produced by keys, in addition to the onset loudness of pistons – thrust and gyration, can be modulated by a combination of twisting and tilting the karlax in this composition. Twisting entails turning the upper and lower half of the instrument in opposite directions. Twisting naturally leads to the performer's forearms and elbows moving away from, or toward, the performer's body. Alternatively, the performer may try to maintain his of her forearms in a rigid position and twist the karlax by bending at the wrists. However, this may

Figure 8. Gradual change from one tablature grid to the next grid.

eventually lead to physical discomfort and stiffness in the wrists as other techniques (e.g., complex fingerings, shaking, stirring) are added and prolonged. Tilting describes a movement in the frontal plane. This means leaning the lower half (i.e., right hand) of the karlax forward and backward from the perspective of the performer. Attention should be given to the lower half because the component for sensing tilt (i.e., accelerometer) is contained within this part of the instrument. The most responsive tilting range encompasses a 180-degree movement from a horizontal position where the karlax is leaning onto its backside (pistons of the right hand pointing upward) to a horizontal position where the karlax is leaning onto its front (pistons of the right hand pointing toward the ground). Increasing volume is achieved by a combination of: (1) twisting "elbows in" – turn the upper and lower half in contrary motion such that the forearms and elbows come toward the body and (2) tilting the instrument backward in the frontal plane. Decreasing volume is achieved by a combination of: (1) twisting "elbows out" – forearms and elbows move away from the body and (2) tilting the instrument forward in the frontal plane.

No unique notation symbol is used in the score to describe twisting-tilting. Rather, the performer must observe traditional symbols for dynamics such as forte, piano, crescendo, etc.. and manoeuvre (i.e., twist and tilt) accordingly.

Practice is required in order to find the most appropriate degree of twist and angle of tilt for specific moments in the composition. Furthermore, the dynamic range of sounds varies according to register/frequency space. For instance, the low register of the left hand, which produces flute-like tones, requires more volume (i.e., stronger twist and leaning the instrument backward) than the high register. The same is particularly true for the uppermost frequency space of the right hand. Ear-piercing tones may be produced if the volume is too high. Consequently, the performer should twist "elbows out" and lean the instrument forward before playing the upper most register of the right hand. This type of response between dynamic range and frequency response is similar to acoustic instruments and, thus, the performer should approach the karlax with a sensitivity to traditional music-making on acoustic instruments.

5.4 Timbre modulation

5.4.1 Tablature grid

The performer will learn how to position his or her instrument in space by reading karlax tablature. In **Figure 8**, the dot within the 5 x 5 grid represents the physical orientation of the karlax, as expressed through tilting in both the frontal and horizontal planes. In other words, the location of the dot is a representation of tilting the instrument for

Figure 9. Sequence of symbols indicating different twist amounts.

ward, backward, to the left and right. The performer should initially follow both the score and the controller patch graphical user interface (in Max) in order to master an understanding of karlax tab. The controller patch provides a visual representation of tilting in the form of a tablature grid, in which the position of the dot is updated in real time. Consequently, learning to tilt the karlax entails matching up the tab in the patch (i.e., on the computer screen) with the tab in the score. With regular practice, the performer will learn to associate tab and karlax orientation in space and the need to look at the computer screen will be eliminated.

Tablature grids are located above the staff lines. Generally speaking, tablature grids are used minimally only as reference points – starting points at which a sound is initiated. Consequently, the performer should not be constrained by maintaining an indicated grid throughout the composition. Freely tilt the instrument in an effort to achieve a unique sound. A dashed line between two grids indicates a gradual change/transition from one karlax orientation to another (see **Figure 8**).

5.4.2 Twisting

Twisting entails turning the upper and lower half of the instrument in opposite directions and naturally leads to the performer's forearms and elbows moving away from, or toward, the performer's body.

A continuum, or sequence, of notational symbols placed above a staff line, is used to indicate twisting. The accompanying graphic illustrates seven possible symbols in the continuum, although more symbols for finer gradations of twisting could be placed with this sequence (see **Figure 9**). The first symbol in the series represents a twist angle in which the elbows are away from the performer's body, while the last symbol represents an angle in which the elbows are near to the body. Moreover, each end of the continuum represents the twist angle before entering the final resistive twist spaces that exist at the maximum degree of twist – maximum torque in either direction. As a point of reference, the third symbol in the continuum corresponds to a twist angle in which both halves of the karlax are in perfect alignment – the lines on the sides of the upper and lower half should be perfectly aligned. Twisting indications are used minimally similarly to tablature grids. Consequently, the performer should not be constrained by maintaining an indicated twist angle throughout the composition. Freely twist the instrument in an effort to achieve a unique sound, especially during sections labelled "ad lib" in the score. A dashed line between two twist symbols indicates a gradual change/transition from one twist angle to another (see **Figure 10**).

Executing a combination gesture of twisting and tilting may be used to control volume, as described above. However, twisting the karlax can also be used in isolation with

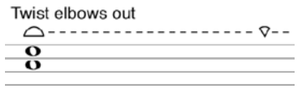


Figure 10. Gradual twisting over time.

right-hand keys. The audible effect will be apparent by practicing this technique.

6. CONCLUSION

In this article, I provided descriptions of several new karlax playing techniques, along with examples of a notation system for conveying these techniques. In particular, I focussed on techniques for: initiating a sound, which includes being able to mute, or silence, a sound; controlling volume; and modulating timbre. All of these techniques are used in my composition, entitled *Ritual*.

With respect to initiating a sound, I illustrated the importance of a musical staff lay-out that reinforces the intrinsic positioning of the left-hand over the right-hand – a contrary orientation was proposed in Mays and Faber.[7] In addition, with respect to using karlax keys, I designed a fingering system that yields higher tones when fewer keys are depressed and lower tones when more keys are held down.

For volume control, I developed a parameter mapping for controlling loudness *via* a holistic gesture combination of twisting and tilting simultaneously. Furthermore, I utilised the traditional crescendo/diminuendo musical symbols and, thus, presented a symbology that is familiar to music practitioners. Mastering volume control entails a holistic interaction while exercising listening skills.

In this article, I described only two performance techniques for modulating timbre. First, I developed a tablature system, which is read both in the musical score and on a computer screen – with practice, the performer may gradually learn to play without looking at a computer screen. The tab "grid" in the score provides an abstract illustration of karlax orientation in space – unlike guitar tab, the symbology is abstract. Instead of providing a graphic that depicts the device, itself, I designed a symbology that parallels the abstract nature of traditional music notation. In this way, the performer remains in a holistic mode of interaction, instead of having to picture the device as an object-technology separate from the performer's body, which was the approach taken by Mays and Faber. The second technique for modulating timbre entails twisting the karlax. Importantly, the custom-designed graphics for twisting also require an awareness of body and, thus, a holistic perception of instrument as an extension of bodily movement is required.

The techniques, mappings and notation described in this article are designed to help the music practitioner understand the music. I describe the experience of playing and listening to the music/sound, instead of describing the experience of manipulating the device. In the NIME community, we must avoid "mixing console" performance techniques. It is my hope that the descriptions provided will help the practitioner play with feeling and escape the console paradigm.

7. REFERENCES

[1] Cadoz, C. (1988). *Instrumental Gesture and Music Composition*. In the proceedings of the 1988 International Computer Music Conference. Cologne, Germany.

[2] Dobrian, C. & D. Koppelman. (2006). *The 'E' in NIME: Musical Expression with New Computer Interfaces*. [In the proceedings of the 6th International Conference on New Interfaces for Musical Expression]. Paris: IRCAM.

[3] Fels, S. S., A. Gadd & A. Mulder. (2002). *Mapping transparency through metaphor: towards more expressive musical instruments*. Organised Sound 7(2), 109-126.

[4] Stewart, D. A. (2015, June 18). *Towards a ritual*. Concert in the University of Lethbridge Recital Hall. <https://vimeo.com/131390417> (2016, February 15).

[5] Iazzetta, Fernando. *Meaning in Musical Gesture, Trends in Gestural Control of Music*. CD- ROM, eds. Marcelo M. Wanderley and Marc Battier. Paris: Ircam—Centre Pompidou, 2000.

[6] Lindström, E. et al. (2003). *"Expressivity comes from within your soul": A questionnaire study of music students' perspectives on expressivity*. Research Studies in Music Education. 20: 23-47.

[7] Mays, T. and F. Faber (2014). *A Notation System for the Karlax Controller*. [In Proceedings of the 2014 Conference on New Interfaces for Musical Expression (NIME14)]. London, UK.

[8] Menuhin, Y. (1996). *Unfinished journey*, London: Methuen.

[9] Minassian, C., Gayford, C., & Sloboda, J. A. (2003). *Optimal experience in musical performance: a survey of young musicians*. Paper presented at the Meeting of the Society of Education, Music and Psychology Research, London, March 2003.

[10] Poepel, C. (2005). *On Interface Expressivity: A Player-Based Study*. In the Proceedings of the 2005 International Computer Music Conference. Vancouver, Canada.

[11] Ridenour, T. (2002). *The Educator's Guide to the Clarinet: A Complete Guide to Teaching and Learning the Clarinet*. Duncanville, Texas, USA: W. Thomas Ridenour.

[12] Roven, J. B. et al. (1997). *Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance*. Presented at "Kansei - The Technology of Emotion" Workshop. Genova, Italy.

[13] Ryan, J. (1991). *Some remarks on musical instrument design at STEIM*. Contemporary Music Review 6(1): 3-17.

[14] Schloss, W. A. & D. A. Jaffe. (1993). *Intelligent Musical Instruments: The Future of Musical Performance or the Demise of the Performer?*. Journal for New Music Research 22(3): 183–193.

[15] Waisvisz, M. (2006). *Enchantment vs. Interaction*. Manifesto delivered at the 2006 Conference on New Interfaces for Musical Expression. Paris: Institut de Recherche et Coordination Acoustique/Musique.

[16] Wessel, D. & M. Wright. (2002). *Problems and Prospects for Intimate Musical Control of Computers*. Computer Music Journal 26(3): 11-22.

Music Industry, Academia and the Public

Carola Boehm
Contemporary Arts, Manchester
Metropolitan University, UK
C.Boehm@mmu.ac.uk

ABSTRACT

What kind of partnerships are best placed to drive innovations in the music sector? Considering the continual appetite for new products and services within our knowledge economy, how can we ensure that the most novel and significant research can be applied in and exploited for the market? How can we ensure that the whole music sector, including the not-for-profit sector, benefits and is engaged in new knowledge production? This paper represents an exploration of a partnership model – the triple and quadruple helix – that is specifically designed to drive innovation. Applying this to the music technology sector, the presentation will provide aspects of case examples relevant for driving innovations in music technology, the creative sector and digital innovations. It will cover both the for-profit sector and social enterprise, and emphasize the importance of partnerships and community for maximizing sustainability when devising research and development projects using helix system models.

1. INTRODUCTION AND BACKGROUND

All universities are involved in partnership work related to their research and enterprise interests. In the area of music technology this may include patenting music instruments, production of music scores, recordings and live performances and researching into new modes of composition and audio production. These activities are often contextualized academically as research and development (gadgets) or practice-as-research (engagement in creative processes). Within my institution's vision statement, we have a section that suggests we engage in 'transformational partnerships'. Like all other universities, we believe we make a real impact on the communities and commercial sectors with which we work.

This is specifically valid for the music sector, which interfaces heavily with external communities, related to cultural assets in forms of concert series, music in the community, music therapy or the music industry.

For academics and creative practitioners in the music technology sector, where subject matter straddles both science and art, technology and creative practice, often involving both commercial and social enterprise, there are questions about how to best to support partnership projects and how to improve the flow from a research stage to the application of these new insights into an external sector.

What makes the consideration of knowledge production in this area even more difficult is that within UK academia, there still seems to be an encultured difference between 'research' and 'enterprise', with the relatively new term on the block being 'knowledge transfer'. Universities may express their intention and policy of treating research and enterprise as a continuum, but just a brief look at career development opportunities within UK institutions, or research quality assurance frameworks, demonstrates a strong preference for basic research over enterprise. This represents a distinct disincentive for academia to engage more directly with industrial partners and/or communities representing end-users. This prioritization of basic research over applied research, or what has been termed as a prioritization of 'Mode 1' research over 'Mode 2 and 3' research, has the potential of slowing the knowledge exchange between academia and industry down, if not stopping it altogether.

Similarly, disincentive models exist in the area of social enterprise, often falling into the category of community engagement, widening participation and/or the 'civic duties' of a university. Many of these terms emphasize the perspective of the educating institution; they are university-centric and are conceptualized as activities that flow within and out of academia. It is this – an increasing number of academics and professionals would argue – which is problematic for forming partnerships that are impactful in allowing research and new knowledge to add significant value both to the sector and to society.

Specifically for music technology, the Higher Education sector divide between research and enterprise has meant that it is difficult for technological innovations coming out of universities to transfer quickly onto the market or external sectors. This difficulty in bringing an idea to the final market stage is perceived to be normal. The external sector thus often perceives universities as too slow to support innovation or to bring an application to market. The supporting structures and incentive models within academia often support the production of journal papers, but the journey from transferring this knowledge to developing a prototype, securing patents, developing market plans, designing for mass production and finally delivering a commercial product is so difficult that too many academically housed music technologists are opting for the traditional publish-a-paper route.

This situation does not need to be this way, and various voices from different sectors suggest that universities need to change the way in which they contextualize, value, incentivize and support research in order for the development of innovation and its application in society to happen much more instantaneously. Authors relevant for this debate are Etzkowitz [1], Carayannis and Campbell [2], Gibbons [3], Watson [4] and Boehm [5] among others, but there is also a wider relevant debate about the role of universities today, including contributors such as Collini [6], Barnett [7, 8], Graham [9] and Williams [10]. The progressive terms relevant for the future are 'triple and quadruple helixes', 'Open Innovation 2.0' and 'Mode 3 research'.

To contextualize this in an example: if we look into the area of assistive music technologies, the market for technologies could be characterized as lacking competition and a consequently lacking diversity and choice. This is in an area where there is still a big end-user need. Supported by the research councils, the area of assistive music technology has always been one with a lively research and development community; many digital innovations are developed for specific special needs communities but they are far less often being turned into commercial products or refined towards mass production. Obviously it will be debatable whether mass produced instruments are as effective in supporting specific communities in need of assistive music technologies, compared to bespoke, uniquely designed instruments for a very particular set of requirements. However, it is exactly this diversity and range - from specifically designed unique instruments to mass produced accessible technologies - that is missing, and it might be argued that this situation is exacerbated by slow research to product introduction channels. The question here is how to combine client-centred approaches, and in-deed 'client co-created solutions' with product development mechanisms to allow the pathways towards introduction of technologies to be shorter than at present, more accessible and more low cost.

These challenges can be overcome more easily by having the right academia–business–government partnerships from the outset of a project, with a more collective and collaborative experience of both basic research and development, as well as application, commercialization and subsequent marketization. Additionally, with the new UK government-driven impact agendas for Higher Education, these issues are timely and relevant to a consideration of the role that universities play in society today. This paper thus focuses on communities, enterprise and the cultural sector involved in, or interacting with, music technological practices, making explicit the various interacting agendas with their respective stakeholders. It attempts to identify ways towards achieving a balance between inward- and outward-facing interests when considering collaborative projects that drive innovation.

The paper will use five main secondary sources: Etzkowitz [1], Watson [4], Carayannis and Campbell [2], Watson [11] and Gibbons [3]. These were written with a general academic perspective in mind, but I will apply the relevant themes in a specific music technology and arts context. The paper will apply these current concepts to innovation developments in music technology, covering both the for-profit and the not-for-profit sector. Providing example projects, I will suggest that triple and quadruple partnerships (e.g. helix models) between universities, industry, government and the civic sector (the not-for-profit and voluntary sectors) allow innovation to happen as a non-linear, collaborative process with overlapping processes of basic research, application and development. In this model, knowledge production (e.g. research) is not the sole concern of universities; and technology exploitation may not be the sole concern of industry; creating what has been called a 'socially distributed knowledge' [3] or a (Mode 3) 'Innovation Ecosystem' [2].

2. RESEARCH AND ENTERPRISE: A PERSONAL EXPERIENCE OF TORN IDENTITIES

Universities are complex and diverse entities. Academics continually live in this 'super-complexity' [7]. They and their academic communities have shifting and changing agendas that – apart from education – allow individuals to engage in research, in enterprise and in community-facing activities. The increase in managerialism, professionalism and centralization has introduced larger amounts of accountability and measurement, and it has followed that activities in the area of enterprise and research are often treated separately, in order to be supported and measured in detail (see also [12]).

A current theme within our knowledge economy is that there are increasing demands on universities to have an impact on society, to interface with the business sector, to commercialize and to be enterprising, while still having supporting structures and incentive models that see civic engagement, enterprise, research and education as very different spheres, supported often by different sections and policies within the same university. Thus the government-driven impact agendas have, probably unexpectedly, resulted in highlighting that the neo-managerialistic cultures with their specific accountability measures are increasingly becoming the barrier to a more holistic consideration of impact – one that exploits the multidirectional benefits of engaging in research, enterprise and civic engagement all at the same time.

I started to consider questions of how best to support collaborative knowledge production and innovation projects a few years ago, when I had to justify yet again why I – an academic at a research-intensive university – was involved in projects that my university at the time classed as *not* research, but 'only' enterprise. I was confident to argue that all these activities produced new knowledge, and all resulted in peer-reviewed journal publications, the classic method for evaluating 'researchiness' in universities. However, there still seemed to be barriers within the university and the Higher Education quality frameworks to valuing something that does not show the classic linear progression from basic research, via dissemination through publications (co-authored in the sciences, single-authored in the arts and humanities), knowledge transfer and application, and external dissemination, to finally having some societal impact.

Similarly, until recently there were plenty of times when I had to argue that several of my projects which included communities and/or businesses were to be defined not only as exclusively 'community outreach' or 'enterprise', but actually as research in action. Even though there were publications as outputs, simply because the funding came from a heritage organization, or a business benefited from the knowledge produced, I seemed to be unable, or able only with difficulty, to col-

lect those brownie points that would allow me to progress on my research-related ladder of academia. The incentives here were geared towards basic research, but not towards impactful community-facing or music-industry-facing product or service development.

This situation is changing fast, and I would suggest that now, after the first dust of the impact debate has settled, there is a real will to make university research (even) more impactful. One of the biggest shifts in the UK that allow universities to consider developing their research cultures into something different is the government's decision to make societal impact a substantial factor for evaluating the quality of research. This is important for universities because of the linked allocation of governmental research funding, now influenced not only by the peer-reviewed and perceived value of the piece of research as evidenced through academic publications, but also by the reach and significance that it has on the external sector, as evidenced through case studies.

Music technology academics have always found it hard to distinguish between technology and artistic practice, enterprise, community outreach and research. One simply has to consider the range of topics and diversity of speakers at the relevant international conferences in this area, such as the International Computer Music Conference, The Art of Record Production Conference or the new Conference for Innovation in Music. Many of the collaborative projects in the area of music technology simultaneously include partners from small and medium-sized businesses, cultural organizations and academia.

To make these developments even more impactful and effective, it is useful to consider partnership models in which knowledge production is not the sole concern of universities, just as technology exploitation may no longer need to be the sole concern of industry. Digital technology and the knowledge economy have allowed the spheres of academia and industry to be shifted, to be realigned. The question is, is this true of the research cultures within Higher Education? With knowledge traditions going back centuries, have they moved with the times, or are they possibly finding it too difficult to keep up with these societal developments? For me, the question emerged of what an ideal engaged *and* entrepreneurial university would look like, and this question involved dealing with understanding and resolving some of the tensions between outward- and inward-facing vested interests, research methodologies and how the quality of research and knowledge transfer is measured.

For each institution, there is the equilibrium of sustainability to be met in an ever-shifting climate of agendas – not a straightforward measurement, considering that the activities are often funded via a complex mixture of sources. This is where an explicit conceptualization of partnerships and vested interests helps.

3. TRIPLE AND QUADRUPLE HELIXES

The triple helix was first described by Etzkowitz in 2008 [1] and provided a conceptual framework for capturing, analysing, devising and making explicit various aspects of project partnerships, 'managing interactions among universities, business and government on common projects'.

The basic assumption of this conceptual model is that in our knowledge-based economy interaction between university, industry and government is key to innovation and growth. In a knowledge economy, universities carrying out research and development become a paramount asset in innovation-intensive production. This can be seen as a historical shift from industrial society, in which the primary institutions were industry and government, to the present knowledge-based society, where economies are much more tightly linked to sources of new knowledge and universities are becoming more important as structures with an everlasting flow of talent and ideas through their PhD and research programmes. Exemplars of this development can be seen in the emergence of university-owned and university-run science parks, incubators, cultural centres and enterprise hubs. Etzkowitz defines it as follows: *'The Triple Helix of university-industry-government relations is an internationally recognized model for understanding entrepreneurship, the changing dynamics of universities, innovation and socio-economic development.'[1]*

Universities in this context of a knowledge economy have the big advantage that they have an inherent regular flow of human capital, such talent and ideas. This is a distinct difference from the research and development sections of large businesses and industry, where the employment structure creates much less dynamics or mobility within its own human capital.

However, in this new economy, the different spheres each also take the role of the other, and there is a much greater overlap of remits and roles than in prior centuries. In this model:

- Universities (traditional role: teaching and learning, human capital, basic research) take the role of industry when they stimulate the development of new businesses through science parks and incubation hubs.
- Businesses (traditional role: place of production, vocational training, venture capital, firm creation) develop training to ever higher levels, acting a little like educational establishments, even universities (e.g. higher apprenticeship schemes).
- Government (traditional role: regulatory activities, basic research and development funding, business support, business innovation) acts often as a public venture capitalist through research grants and studentships, including, for instance, knowledge transfer partnerships.

This overlapping of the formerly distinct roles of three different spheres (in the case of the triple helix) suggests that the traditional stages of knowledge transfer from

- Stage 1 government – university (example: research grant)
- Stage 2: university – business (example: incubator)
- Stage 2: government – business (example: business start-up grant)

overlap much more, and more often, than they have done traditionally.

Etzkowitz's model was expanded in 2012 by Carayannis and Campbell to include the third sector, and with it universities' own civic engagements. Watson [4, 11, 13] has foregrounded this latter role; his concept of the 'engaged university' proposes that social enterprise and the not-for-profit sector should be considered within the helix model. His international comparison of the way universities engage with their respective communities provides a strong articulation for academia to consider new knowledge production models that allow a greater interaction between universities on the one hand and both the public and industry on the other, for example for universities to become (even?) more engaged.

Various arts-related initiatives have attempted to use these models to initiate innovation [14, 15]. Similarly, because of their inherent use of inter-, multi- and transdisciplinary knowledge production methods, the potential that helix partnerships provide for managing large-scale and multi-partner projects allow these concepts to come to the fore in considerations of the world's largest challenges. Addressing its impact potential on the socio-economic aspects, Watson suggested that in this new era universities have to become more 'engaged', and he specifically points his finger at universities in the northern hemisphere [4].

At the core of this debate stands the notion that our classic (northern hemisphere) research methodologies and their related cultures, frameworks and value systems are preventing us from increasing the impact on society. Universities that value socio-economic impact will thus always have an emphasis on partnerships between universities, industry, government and the civic sector (the not-for-profit and voluntary sectors).

Not only will these quadruple partnerships better support innovation, but they will allow innovation to happen in a non-linear, collaborative manner with overlapping processes of basic research, application and development. In this model research is not the sole concern of universities, and technology exploitation may be not the sole concern of industry, creating what has been called a 'socially distributed knowledge' [3] or a (Mode 3) 'Innovation Ecosystem' [2].

These debates feed into an ever-increasing discourse around the comparative appropriateness of various research methodologies for benefiting the real-life problems of society, from inter-disciplinary or trans-disciplinary methodological considerations to practice-as-research [16] and the creative practitioner; from the challenges of big, co-owned and open data or non-linear collaborative methods for producing knowledge.

What have given a renewed focus on how academia interfaces with communities outside of itself, allowing the Higher Education sector to produce knowledge that has real impact, are the last Research Excellence Framework (REF) in 2014 and the government-driven agendas concerning impact. The last REF could be seen as a collection of quality assessment methods that collectively have an inbuilt tension of, on the one hand, a more traditional, linear knowledge production culture (Gibbons's Mode 1 knowledge production model) and, on the other, an impact-driven, non-linear mode that values socially distributed knowledge more than discovery (Gibbons's Mode 2 knowledge production model) [5].

4. GIBBONS, CARAYANNIS AND CAMPBELL AND THEIR KNOWLEDGE PRODUCTION MODELS

Mode 1 and Mode 2 were knowledge production models put forward by Gibbons back in 1994. Several authors of the past decade have picked up and further developed his concepts with relevance for the current impact agendas. The relevant works include Etzkowitz's 'The triple helix' [1], Watson's *The engaged university* [4], Carayannis and Campbell's *Mode 3 knowledge production* [2] and Watson's *The question of conscience* [11].

Gibbons conjectured that Mode 1 knowledge production was a more 'elderly linear concept of innovation', in which there is a focus on basic research 'discoveries' within a discipline, and where the main interest is derived from delivering comprehensive explanations of the world. There is a 'disciplinary logic', and these knowledge production models are usually not concerned with application or problem solving for society. Quality is primarily controlled through disciplinary peers or peer reviews; Carayannis and Campbell add that these act as strong gate keepers. Success in this model is defined as quality of research, or 'research excellence' and both Watson [4] and Carayannis and Campbell [2] suggest that our Western academic cultures still predominantly support the Mode 1 knowledge production model. The REF's focus on scholarly publication and its re-branding to include the term 'research excellence' may be considered as emerging from a culture surrounding the traditional Mode 1 knowledge production.

But Gibbons had already put forward a different way of producing knowledge, in which problem solving is organized around a particular application. He suggests that the characteristics of this mode are greater inter-, trans- and multi-disciplinarity, often demanding social accountability and reflexivity. The exploitation of knowledge in this model demands participation in the knowledge production process; and the different phases of research are non-linear, for example discovery, application and fabrication overlap. In this model, knowledge production becomes diffused throughout society for instance a 'socially distributed knowledge', and within this, tacit knowledge is as valid or relevant as codified knowledge [3]. Quality control is exercised by a community of practitioners 'that do not follow the structure of an institutional logic of academic disciplines' [3], and success is defined in terms of efficiency and usefulness in contributing to the overall solution of a problem [17]. Mode 2 is seen as a natural development within a knowledge economy, as it requires digital and IT awareness and a widely accessible Higher Education system. Research cultures using Mode 2 models often initiate a greater sensitivity of impact of knowledge on society and economy.

Obviously, the two modes currently exist simultaneously in various research communities, and have done so for a long time. Various terms emphasize the different nuances of the ongoing impact debate, from applied research, through knowledge exchange, to definitions of

research impact. However, as Watson [4] contends, there is a distinct divide between the southern and northern hemispheres in how academia tends to see itself and its role in relation to society, and embedded in this is how research value is conceptualized.

In the northern hemisphere academia generally comes from a Mode 1 trajectory, that is, Mode 1 knowledge production is, more often than not, considered to be the highest form of research. This is reinforced by publicly funded research that creates a sense of entitlement [4], and generally there is more panic about the decline of interest in scientific and technological study, with many degrees being kept alive by students from overseas. For universities in the northern hemisphere, Watson's list of characteristics includes the following:

- They derive much of their moral power from simply 'being there'.
- They are aware of their influence as large players in civil society.
- They stress role in developing character and democratic instincts.
- They focus on contributions like service learning and volunteering.
- They see public support for the above as an entitlement.
- The main model of contribution is knowledge transfer.
- They have developed from a culture in which Mode 1 is valued as the highest form of research.

This cultural stance can also be detected in the role that universities play as cultural patrons. There is a sense that art is entitled to public funding, and there is a long history of publicly funded art – specifically in the UK.

For universities in the southern hemisphere, civic engagement is an imperative, not an optional extra. Watson writes that in his team's enquires, 'we were constantly struck in our Southern cases, by how much was being done by universities for the community with so little resources (and with relatively little complaint)' [4].

'Practical subjects' and 'applied' research take priority and with them comes a different value system for the role of research: the Mode 2 knowledge production model prevails [3, 4]. Thus Watson sees Mode 2 as a more progressive developmental stage of Higher Education in reference to societal impact and civic engagement. His list of characteristics includes:

- It simply is more dangerous – there is no comfort zone.
- There is an acceptance that religion and sciences should work in harmony.
- There is a general use of private bodies for public purposes.
- International partnerships are for assistance, not 'positioning'
- Challenging environments[1] where many attacks on universities seem to be connected to various governments' efforts to prevent opposition movements, restrict political debate or criticism of policies [18].
- There is frequently a central political drive for outcomes like 'transformation' (South Africa) or 'solidarity' (Latin America) (Leibowitz 2014:47).
- There is a privileging of 'development' (or social return) over 'character' (and individual return), of 'national cohesion' over 'personal enrichment'; and of 'employment' (human capital) over 'employability' (SETs (Science-Engineering-Technology) over arts).
- International partnerships are there for assistants, not 'positioning'.
- 'Above all "being there" doesn't cut much ice; there is a much greater sense of societal pull over institutional push' [4].

Thus, there is a predominant engagement with Mode 2 knowledge production.

In 2012 Carayannis and Campbell expanded the concept of Modes 1 and 2 to include a Mode 3 knowledge production model, defined as working simultaneously across Modes 1 and 2. Adaptable to current problem contexts, it allows the co-evolution of different knowledge and innovation modes. The authors called it a 'Mode 3 Innovation Ecosystem' which allows 'GloCal' multi-level knowledge and innovation systems with local meaning but global reach. This values individual scholarly contributions less, and rather puts an emphasis on clusters and networks, which often stand in 'co-opetition', defined as a balance of both cooperation and competition.

5. CASE STUDIES

A large set of case studies of helix partnerships related to digital arts innovation was published in a project report in 2014 [14]. CATH – Collaborative arts triple helix was a AHRC funded project between 2013 and 2014 that specifically tried out the triple helix model for digital arts innovations. In the project, they identified barriers and measurable benefits as:

Barriers

- Language and Trust
- The need to define roles
- Commercial concerns, specifically for non-academic partners
- Inflexible academic administrative systems

Benefits

- Access to HE research
- Conducting research, specifically for HE staff
- Reputational gains, non-academic partners
- Access to technical expertise
- Improved problem solving
- Development of future grant applications building on further triple helix models
- Development of new products, prototypes or business models

The AHRC project focused on 'brokered' triplets as a partnership model, and it specifically allowed new partnerships to emerge facilitating innovation by bringing together new sets of expertise and resources.

For the project design mentioned in the examples below, we were considering the quadruple helix model as a framework, and not so much focusing on 'brokerage' as in co-creation. For the music sector, there are various opportunities that a more structured quadruple helix partnership approach can seize. Two research areas can act as examples of how Mode 3 thinking and a helix partnership approach benefit all the sectors involved – the music industry, the public, academia and government – with its societal and economic imperatives.

5.1 Example 1: Hard and Software Developments and Assistive Music Technologies

Music Technology is taught in the UK in various departments, according to UCAS by 103 providers to be exact, with more than 200 degrees situated somewhere within and between the disciplines of Computer Science, Electrical Engineering and the arts. Innovation happens in all of these, and specifically the more 'gadgety' type of innovation often needs industry-related experience and a knowledge of developing products from an idea to a mass-produced item for sale. Although in general Electrical Engineering and Computer Science departments have still more experience in these processes than arts and humanities departments, even here there are barriers that do not always allow good ideas to be developed into products. In view of the fact that our new knowledge economy needs *more* products, a more *diverse* range of products and *cheaper* products, the pathways from initial research to product really do need to be shortened. The industry sector is geared up for this, and modern innovations such as 3D printing and rapid prototyping have made the production of diversity in product development cheaper than it ever was before.

In fact, there have been plenty of individual instrument developments as part PhD studies and funded research projects; but of these, only the smallest number of ideas and prototypes have been developed towards industry exploitation. Plenty of examples exist where a prototype represents the final stage of the research project, and the lack of collaboration and/or incentives for individuals to develop it to marketization, as well as a real lack of incentive models within institutions, keep the knowledge just there with the individual. This individual often stays within academia, and is thus able to gain career advantages not by marketization, but by publication of the idea and concept. This may still be seen as a classic form of the ivory tower. Thus for the area of instruments or gadgets for special needs musicians, there is a distinct need to shorten the pathways from university research to market availability.

As one solution, we have been developing projects based on the quadruple helix model and a Mode 3 research methodology. In it we aim to connect the relevant communities with the micro and SME (Small and Medium Enterprise) market, supported by innovations derived from university research by PhD students and academics. The idea is for us academics to collaborate on developing a new series of digital innovations, together with end-users and SME developers. Thus the knowledge will not be located only within the Higher Education institution, but will be shared among the partnership, and – importantly – between SME and Higher Education.

In Gibbons's terms, knowledge will thus be (more) socially distributed in this non-linear model, and discovery, application and fabrication will overlap. The control of quality will be exercised by the community of practitioners who (and I quote Gibbons again) 'do not follow the structure of an institutional logic of academic disciplines' [3]. These disciplines should not be relevant for evaluating the quality and success, as this is not defined by the Mode 1 model in terms of excellence (evaluated by peer review), but by Mode 3 models and in terms of efficiency, usefulness and contribution to an overall solution to a problem.

Obviously, university structures still tend to show some friction with these new conceptualizations of research and how to value it. But unless we want Europe to continue to fall behind in entrepreneurial and innovative activities, universities will need to find new ways in which to support and incentivize academics in a Mode 3 research model, in order to boost the economy of our knowledge society through real innovation based on knowledge production.

In practical terms, carrying out these helix partnerships in a university context that might still afford a kind of Mode 1 'academic behaviour' can be hard work. It means

- Understanding collaboration to signify co-creation, co-ownership, and possibly multi-professional working. Trust is an important aspect of this. For university academics, this means that university structures will have to be challenged, and time invested for negotiating solutions in the area of intellectual property, grant sharing and income sharing. All this is possible, but depending on institutional cultures and policies, more or less effort is needed to accomplish the starting frameworks in which these types of projects can then happen.
- Funding: many funding streams still differentiate between research or enterprise, and grant structures often insufficiently incentivise SMEs, professionals or community organisations to invest the time into these projects. Projects might be perceived to be too 'research' or academically focussed. Substantial effort (and awareness) is needed to devise project benefits for all concerned.
- Mediation: the need to mediate, interface or broker between and amongst different partners is a substantial additional project management task, and this ideally needs to be costed into the projects when applying for grants.
- Language has been pointed out as being a real barrier to collaboration. Terms come with associated meaning and connotations, and it helps to speak the various sector's language and be able to mediate be-

[1] For example, 2012 northern Nigeria, Federal Polytechnic in Mubi, 46 students killed, pretext student union election. 2013 Nigeria, gunmen killed at least 50 students. 2013 Syria, University of Aleppo, 82 students killed. 2014 Ethiopia, a bomb killed 1 and injured more than 70. 2015 Kenia, Nairobi, Somali militants burst into a university in eastern Kenya on Thursday and killed nearly 150 students. For a full report see Global Coalition to Protect Education from Attack, 'Education under Attack 2014', GCPEA, New York 2014. http://protectingeducation.org Last accessed 09/05/2015.

tween them and make differences in understanding explicit.
- Roles: need to be re-defined between partners as part of the project, but also defined in intra-institutional terms.

All these aspects are negotiable, but need time investment, so the benefits need to be understood and made explicit from the start.

5.2 Example 2: Music- and Arts-Related Multi-Professional Work (MPW)

Similarly, in another European project we are developing training packages for multi-professional or inter-agency community arts and community music workers. This project is simultaneously a community arts project in itself and a project to define and develop new multi-professional working skills and environments for professionals in art and social work.

Music of course has a big potential for engaging with external communities, whether it is in the context of being a cultural asset (concert series), a creative practice (music production, audio engineering, composition, performance), music therapy (assistive music technologies), music technology (plugins, apps), or simply being an anchor for economic regional growth and supporting new talent from all areas of the music industry and the creative sector.

In this project, however, the *new* knowledge (the definition and identification of skills and competencies in an inter-agency or multi-professional community arts setting) is gained within a partnership model that includes lecturers, representing academia; artists, representing the creative sector; end-users, represnting community; and the European Commission, representing the governmental part of the helix.

It is no wonder that this was always likely to be a Creative Europe or Erasmus+ funded project, and not a Horizon 2020 project. Creative Europe and Erasmus+, with their inter-cultural and socio-economic missions, are perceived to be a more appropriate funding body to target projects that use Mode 3 research, as their activities and outputs are still considered more under the headings of community outreach, cultural work, education and/or enterprise.

However, even Bror Salmelin [19], a director-general of the European Commission, who presented at a European conference in Finland recently, emphasized the need for the European research community to embrace Open Innovation 2.0 models, including quadruple helix thinking. Quadruple Helix Models supports nicely the MPW nature of this project, as the emphasis of this kind of MPW work lies on dividing work between professionals while working together with young people and on the definition of genuine MPW cooperation and collaboration. It is a MPW practice stemming from a multidisciplinary approach to working with communities and individuals. As the initial project documentation suggests, there are "… artists who are willing to work in new kinds of environments. In the field of social work there is a growing will to apply art, but it is not always easy when different professional cultures confront". [20] Artists might feel that they cannot get inside the community of social work professionals or might perceive that by doing so, they leave their artistic integrity behind. Social Work/Care professionals, on the other hand, often feel that collaboration complicates their work, and there is often a lack of confidence in applying artistically informed approaches. More often than not is there real enthusiasm and willingness, but perceiving themselves not as artists, the validity of how they use artistic methods, its artistic integrity, is perceived to be associated with a deeply informed, embodied and/or studied practice and thus represents a barrier towards a more wider, more common or deeper application of arts approaches in social work/care contexts.

The triple helix system allowed us to try out models of co-creation, co-ownership and collabroation whilst developing new educational frameworks that would facilitate new multiprofessional skills and comptencies.

This project is into its first year, but the model has already manifested itself in multi-named and co-authored articles [21], practices that were shared across the whole partnership (and in 4 European Countries) and most improtantly for the search of new knowledge and innovative practices, a deeper understanding of the terms and meanings associated with arts, health and wellbeing and their specific national contexts and the imlications for effective training models to suuport these contexts.

6. CONCLUSION AND WAYS FORWARD

Bearing in mind Watson's suggestion that in the north we tend to engage predominantly in Mode 1 research (in contrast to the south's Mode 2), and thus are consequently somewhat less engaged in partnerships that could be considered triple, quadruple or even quintuple [2] helix models, it may be worthwhile to consider that even in the north, partnership work in publically funded research has been the norm. Thus, although they are not consciously implemented or explicitly formulated in policy, project parameters that conform to helix models can be identified extensively.

The concept itself, however, gives us various opportunities that have yet to be explored more widely, specifically in the music industry and cultural sector. The model has been evidenced to enhance innovation, and with the reduction of funding for the arts, universities – with their large sustainable amount of human capital – must increasingly become the place of viable patronage. Partnership models are thus increasingly important. The model also allows industry to have access to Higher Education research, without the more lengthy traditional routes of research – knowledge transfer – commercialization. In this model, the whole partnership will be (more or less) engaged in the research process, as well as in the commercialization. Where models have been adapted in other commercial sectors, the path to market has been shortened [14].

Project partnerships that have engaged in helix models report a better knowledge exchange and more effective partnership work for securing further funding to develop additional products. Helix partnerships help sustainable collaborations to emerge [14]. Finally, the powerful conceptual framework allows us to leverage stronger policy around research funding – allowing Mode 3 research partnerships to become more the norm and thus maximizing impact. Implicit examples for these can be seen in the EU's Creative Europe Programme.

The explicitness of the model allows the capture, analysis, reflection and explicit making of various aspects of project partnership work. With these in place, project interactions between universities, business, public and government can be managed in a rigorous framework of relationships.

With the realization that universities need to engage more, as evidenced by the current impact agendas within academia, and to maximize their impact of their own research, the debate on how to foster partnerships that more effectively turn new knowledge into benefits for industry and society has begun. Helix partnerships, Mode 3 research models and Open Innovation 2.0 are the concepts that are currently considered to be a solution.

For the music industry, if the UK wants to exploit the talent and creativity it has within its midst, partnership work between SMEs, academia and the public is essential. Mode 3 research and triple and quadruple helix structures for partnerships are the best way forward.

7. REFERENCES

[1] Etzkowitz, H. The triple helix: university-industry-government innovation in action. Routledge, New York (2008).

[2] Carayannis, E. G. and Campbell, D. F. J. Mode 3 knowledge production in quadruple helix innovation systems: 21st-century democracy, innovation, and entrepreneurship for development. Springer, New York and London (2012).

[3] Gibbons, M. The new production of knowledge: the dynamics of science and research in contemporary societies. SAGE Publications, London and Thousand Oaks, CA (1994).

[4] Watson, D. The engaged university: international perspectives on civic engagement. Routledge, New York (2011).

[5] Boehm, C. 'Engaged universities, Mode 3 knowledge production and the Impact Agendas of the REF' in Next steps for the Research Excellence Framework. Higher Education Forum. S. Radford, ed. Westminster Forum Projects, London (2015).

[6] Collini, S. What are universities for? Penguin, London and New York (2012).

[7] Barnett, R. Realizing the university in an age of supercomplexity. Buckingham. Society for Research into Higher Education and Open University Press, Philadelphia, PA (2000).

[8] Barnett, R. Reshaping the university: new relationships between research, scholarship and teaching. Society for Research into Higher Education and Open University Press, Maidenhead (2005).

[9] Graham, G. Universities: the recovery of an idea, 1st edition. Imprint Academic, Thorverton, England, and Charlottesville, VA (2002).

[10] Williams, G. L. The enterprising university: reform, excellence, and equity. Society for Research into Higher Education and Open University Press, Buckingham (2003).

[11] Watson, D. The question of conscience: higher education and personal responsibility. Institute of Education Press, London (2014).

[12] Deem, R., Hillyard, S. and Reed, M. I. Knowledge, higher education, and the new managerialism: the changing management of UK universities. Oxford University Press, Oxford and New York (2007).

[13] Watson, D. The question of morale: managing happiness and unhappiness in university life. McGraw-Hill, Maidenhead (2009).

[14] R. Clay, R., Latchem, J., Parry, R. and Ratnaraja, L. 'Report of CATH Collaborative Arts Triple Helix' (2015).

[15] Carayannis, E. G., and Campbell, D. F. J. 'Developed democracies versus emerging autocracies: arts, democracy, and innovation in quadruple helix innovation systems'. Journal of Innovation and Entrepreneurship, Vol. 3, p. 23 (2014).

[16] Linden, J. 'The monster in our midst: the materialisation of practice as research in the British Academy'. PhD thesis, Department of Contemporary Arts, Manchester Metropolitan University, Manchester (2012).

[17] Carayannis, E. G. Sustainable policy applications for social ecology and development. Information Science Reference, Hershey, PA (2012).

[18] G. C. t. P. E. f. Attack, "Education under Attack 2014," GCPEA, New York 2014.

[19] Curley, M. and B. Salmelin, B. "Open Innovation 2.0: a new paradigm'. 2015.

[20] Tonteri, A. Developing Multiprofessional Working Skills in Art and Social Work. (2013)

[21] Boehm, C., Lilja-Viherlampi, L., Linnossuo, O., McLaughlin, H., Kivelä, S., Nurmi, K., Viljanen, R., Gibson, J., Gomez, E., Mercado, E., Martinez, O., 'Contexts and Approaches to Multiprofessional Working in Arts and Social Care', Journal of Finnish Universities of Applied Sciences, Special EAPRIL Issue, Turku, (2016).

A cross-genres (ec)static perspective on contemporary experimental music

Riccardo Wanke
Centre of Musical Sociology and Aesthetic Study – CESEM,
University 'Nova' of Lisbon. Faculdade de Ciências Sociais e Humanas,
Av. de Berna, 26 C, 1069-061 Lisbon, Portugal.
riccardowanke@gmail.com

ABSTRACT

This paper presents a particular perspective, shared across various currents of today's music that focuses on sound itself as a complex entity. Through the analysis of certain fundamental musical elements and sonic characteristics, this study explores a new method for comparing different genres of music characterized by a similar approach to sound. Taking benefit of theoretical and perceptual examinations, this strategy is applied to post-spectralist and minimalist compositions (e.g. G. F. Haas, B. Lang R. Nova, G. Verrando), as well as glitch, electronic and basic-channel style pieces (Pan Sonic, R. Ikeda, Raime). Nine musical attributes are identified that help trace a new outlook on various genres of music. The study's contribution lies in its revealing of a shared musical perspective between different artistic practices, and in the establishment of new connections between pieces that belong to unrelated contexts.

Keeping with the topic of the conference, this paper attempts to deal with several questions, such as (i) the "splendid of isolation" of genres of experimental music, (ii) the development of new cross-cultural methods of analysis and (iii) the future of music education and didactic approaches.

1. INTRODUCTION

The field of contemporary experimental music, considered in its broad sense, is enormously diversified [1, 2], but distant genres of music have in common some characteristics even if they are part of distant cultural and social environments.

Throughout the 20th century, certain currents within exploratory music can be seen as moving progressively towards a more explicit interest in the intrinsic properties of sound [3]. Within this frame of reference, one can identify that starting from the mid-twentieth century certain trends (*i.e.* non-teleological and acousmatic perspectives; the fusion of electronic, acoustic and concrete sounds; and the extended use of sound spectra) were simultaneously developed and established as the cardinal principles of artistic practice across distant genres of music [1].

More recently, spectralism and the exploration of sound using electronic technology have acted as a sort of springboard for the development of new musical genres, namely in the electroacoustic field. At the same time, during the '80s and '90s, there was an on-going process of constant and discrete refinement within many genres of popular and alternative music towards more advanced and sophisticated forms, *e.g.* experimental rock, drone metal, basic channel style, IDM, among others [1, 3, 4]. This process of sophistication within different genres of popular music, often accompanied by more specialized though smaller audiences, lead to a shift of perspective: from music as entertainment and distraction to being something deserving contemplative home listening focused on the sonic experience.[3, 4].

Currently, these different areas of musical exploration consider similar sonic materials and arrive sometimes at equivalent results. Common practices are found across distant styles: such as a flexible approach to harmony, the enormous extension of timbric range, the creative use of new technologies, and, above all, the sculptural approach to sound as a matter to mould. This paper looks at a cross-section of contemporary music and examines their use of fundamental musical elements in order to highlight this shared perspective across different genres of exploratory music.

2. AREA OF STUDY

This study considers those genres that approach sound as a sculptural material, considering it as a complex, dense and tangible entity. This description is admittedly fairly generic and vague but allows going beyond a specific instrumentation: as this paper is drawn to identify common approaches to sound irrespective of the medium used, musical practices are considered whether they are electroacoustic, purely acoustic or electronic.

On the one hand, contemporary composers such as Georg Friedrich Haas, Fausto Romitelli and Bernhard Lang have advanced their artistic practice by exploring new possibilities within instrumental music, each according to his own aesthetic, considering sound as a complex, almost tactile substance. Another characteristic common to these composers is their unconventional vision of time in music. Through their particular use of repetition and their exploration of sound spectra, their work induces a kind of temporal dilation: sound is treated as an almost atemporal object, periodic, cyclic and static.

On the other hand, post-minimalist composers and electronic performers such as Alvin Lucier, Eliane Radigue and Alva Noto have made free use of the musical theories of the "experimental school" of Cage, Feldman and Schaeffer, absorbing these influences to create more "instinctive" works. In some cases, their methods combine noise and tonal melodies (C. Fennesz), granular and digital processes (R. Ikeda), hypnotic repetitive clusters (minimal-techno and basic channel styles), immersive multi-channels soundscapes (B. Labelle), and exploratory sound resonances (J. Kirkegaard). These approaches maintain certain common features such as non-narrative development and a particular focus on the perceptual aspects of music.

Indeed, both of these two broad musical currents have the shared desire '[to] create works that seek to engage the listener in a stimulating listening experience' [5]. This listening experience is characterized by a new vision of time in music, space (i.e. multi-channels diffusion and sculptural musical design), musical evolution (non-narrative and extended) and repetition (generation of hypnotic effects and a listening "in accumulation"). These characteristics form an *(ec)static listening environment*, where the musical material is *static* (atemporal, non-narrative) and the listening attitude is *ecstatic* (free to explore and move through the dimensions of sound) [6, 7].

The variety of the cross-genres area makes a useful comparison difficult: how is it possible to examine the complex composition for 24 instruments *in vain* by Haas together with the indefinite drone of *Kesto#4* by the electronic duo Pan Sonic? In my recent paper [6], I try to approach at this musical material focusing the attention on the primal musical elements and the current study seeks to approach different genres from the side of perceived musical components rather than through comparison of any writing method or theory. Current methods approaching analysis from perceptual aspects are mostly applied to electroacoustic music [8] and focus attention on the nature of sound within a specific genre [9]. However, an effort to extend these approaches to a more general level capable of comparative studies across different styles is currently a subject of great interest among scholars [10], and is the aim of the study proposed here.

3. ANALYTIC PROCEDURE

3.1 Method

The first step of this study is the definition of general categories to classify those musical events that seem typical of our area of study. This framework should consist of fundamental guidelines that can be adapted to different styles (e.g. acoustic, electroacoustic, electronic…) and have the potential to describe the characteristics of sound itself.

Thus, each piece is divided up according to either a narrative partitioning or a separation based on musical textures or events [8]. Each episode is analyzed and described using a combination of four categories: *time*, *dynamics*, *spectrum* and *mode*. The second step looks at the sonic effects of each musical episode, and how these define a particular musical environment (Figure 1, steps 1 and 2).

The framework put forward in this paper is designed to be flexible and comprehensive, taking the characteristic of sound itself to stimulate the listening experience as its starting point. The analytic procedure presented here is best regarded more as a comparative method than simply a taxonomic description. Therefore, this study aims to provide a qualitative classification method with which to identify comparable musical events in a list of compositions. This analytical process was applied to a set of compositions and revealed a large number of similarities that then led to the definition of nine musical attributes.

3.2 Selection Of Pieces

For the purposes of this study, compositions characterized by an absence of overt sociocultural references and containing only a small number of real-world sounds were opted for. Musical works containing narrative voices or representational musical elements would have been ill-suited to the premise of starting with sound itself: extra-musical traits are more often related to social and cultural themes than to the sonic characteristics of a piece. However, the background and context for each piece should be examined [10], allowing the influence of a composer's poietic intention to be acknowledged. Thus, this approach should be able to decode and interpret diverse musical works and be able to identify similarities among them.

The following pieces were analysed: • G. F. Haas String *Quartet nº2* and *In vain*; • B. Lang *Differenz/Wiederholung* (selection); • R. Nova *Eleven*; • G. Verrando *Dulle Griet*, *Triptych #2*; • Pan Sonic *Kesto*; • Ryoji Ikeda *+/-* ; • Raime *If Anywhere Was Here He Would Know Where We Are*, *Quarter Turns Over A Living Line*, and *Hennail*.

This selection reflects a great variety of styles but all pieces display a common focus on the precise use of various spectral characteristics of sound. The range of styles represented allows this study to demonstrate that the common perspective that emerges goes beyond any specific instrumentation or genre as the pieces cover acoustic (i.e. Haas and Lang), electronic (i.e. Ikeda, Raime, Pan Sonic), electroacoustic, mixed and real-world sounds (i.e. Nova, Verrando, Lang and Pan Sonic), and span from contemporary classical music to experimental alternative music genres.

Haas's pieces represent prime examples of contemporary instrumental music that explore various sonic characteristics through a minute exploration of instrumental spectra and a large use of microtonality. Highlighting this approach, *in vain* (2000) includes lighting instructions for live performance, denoting shifts from darkness to fully illuminated, driving the audience's attention towards the perceptual aspects of sound.

The B. Lang project, *Differenz / Wiederholung*, is characterized by the exploration of repetitive musical elements such as looping and the idea of erratic reiteration, suggesting connections with DJ and glitch aesthetics [11]. Verrando's and Nova's pieces combine acoustic instruments, electronic sounds, and noise to explore new limits in electroacoustic composition, embracing, for instance, enharmonic exploration and digital manipulation.

The Finnish duo Pan Sonic (Mika Vainio and Ilpo Väisänen) and the Japanese sound artist Ryoji Ikeda are representative figures of glitch music. The former are closer to an experimental industrial aesthetic and mostly work with analogue electronic devices, while Ikeda usually creates his sound digitally using a computer. Their music is often focused on "raw" elements of sound, such as sine tones and noise, and uses an extreme spectral range pushing at the edges of human hearing. The London-based duo Raime (Joe Andrews and Tom Halstead) are typical of the recent underground scene in that they move freely between noise, techno and dub styles. Their music usually combines the asymmetrical rhythms of dubstep with minimal musical textures.

The audio, spectral and score examinations previously described reveal nine musical attributes that are to a large extent common to all these pieces (Figure 1, step 3). These attributes depict a frame of musical practices that comprises an extended spectral vocabulary, a clear use of repetitive musical units for specific purposes and a peculiar idea of time and space within the sound.

3.3 Nine Common Attributes

These attributes are (Figure 1, step 3):

• (A) An Expanded Spectrum, *i.e.* the use of an extended frequency range. This trait is especially prominent in electronic or electroacoustic pieces. However, this feature is also apparent in contemporary acoustic compositions using traditional instruments: Haas, for instance, makes frequent use of *sforzato* and *sul ponticello* techniques to generate multiple overtones in his string quartet.

• (B) Microtonal Variations, *i.e.* the use of microtonality or more generally interactions between neighbouring frequencies. Specifically, this attribute concerns, (i) the exploration of binaural beats (*e.g.* Ikeda, Haas), and (ii) the creation of static blocks with minimal fluctuation (*e.g.* Pan Sonic, Lang), and is found to a similar degree across the entire selection of pieces.

• (C) Systematic Glissandi, *i.e.* the use of glissandi embedded within repetitive units. This feature is present in most pieces: in the case of contemporary instrumental compositions, it often arises as a core structure used to create larger patterns (Haas, Lang), while in electronic pieces it is used more as a systemic contour to shape continuous evolutions.

• (D) Rhythmic Developments are integral to glitch or techno genres, but occasionally appear as minimal evolution in other contemporary pieces. They are frequently used by Lang within repetitive clusters (F) (see below), but also by Haas who creates synchronous progressions (*e.g.* the final part of *in vain*) that generate a kind of rhythm. In case of styles that are based on pulsations, the rhythmic development should be considered and understood within the specific socio-cultural context of these genres (*e.g.* IDM). Nevertheless, these styles do develop sophisticated rhythmic textures that are associated with timbric, spatial and dynamic attributes identified by this analytical method. In the interests of variety, the pieces selected for this study by these authors also include ones not primarily built on pulsation or rhythmic development.

• (E) Static Masses are constituted by layering sounds, but can also be created through continuous stationary textures, *e.g.* Lang's pieces. As mentioned above, this static attribute embodies an atypical vision of time. The creation of non-teleological musical sections is a common tendency among post-spectralist and minimalist pieces using an immobilized temporal flow to explore sonic nuances. This attitude is also typical of those genres that use drones (*i.e.* the use of sustained, repeated sounds, or tone-clusters) to originate hypnotic effects, such as in the work of Pan Sonic and Raime.

• (F) Repetitive Clusters, *i.e.* unvaried musical motifs that could generate rhythmic patterns (D), hypnotic effects (H), or mechanical and automated profiles. This attribute is strikingly prevalent in the pieces by Haas and Lang, where it is often the building block for more complex musical organizations. Each cluster, as a musical unit, can be composed differently: in the case of *in vain*, for example, the cluster consists of sequential (during the first part) and layered (during the final part) groups of tones. The use of repetitive clusters is also representative of glitch and electronic genres.

• (G) Dynamic Contrasts, *i.e.* the opposition of different elements according to certain sonic dimensions, are usually related to the sculptural use of sound (I) that is the combination and succession of events creating repetition or difference. This attribute is carefully evaluated, more through a qualitative and aesthetic assessment than a quantitative analysis using absolute values. A hasty comparison of the use of dynamic contrast in Ikeda and Haas pieces, for example, could lead to antithetical conclusions. The former makes use of extreme spectral ranges, whereas the latter creates contrasts only within a traditional acoustic palette. However, taking the possible dynamic range of instrumental composition into account, Haas' designs can be seen as equally radical and contrasting.

• (H) Hypnotic Reiterations, *i.e.* repetitive musical elements used both for static (E) and rhythmic development (D) purposes. This dual purpose in creating hypnotic effects using sustained sounds or streams of short tones (E) and continuous pulsations (D) is common to all the pieces selected and helps reinforce the idea that a shared perspective arises from the use of similar practices.

• (I) A Plastic and Sculptural Arrangement of Sound, *i.e.* the use of a particular organization of sounds based on their various characters, be it according to their sonic and spectral characteristics. This attribute, typical of electroacoustic and mixed music, is predominant in those genres that encompass the use of multichannel sound diffusion. In our selection, the sculptural arrangement of sonic elements is focused on the organization of musical events within the inner sonic space in order to create virtual planes and dimensions of perception (*i.e.* Smalley's spectral space and spatiomorphology [10]). In some cases (*i.e.* Pan Sonic and Raime pieces), it is also accompanied by the use of real-world sounds that serve to clearly characterize specific spectral regions.

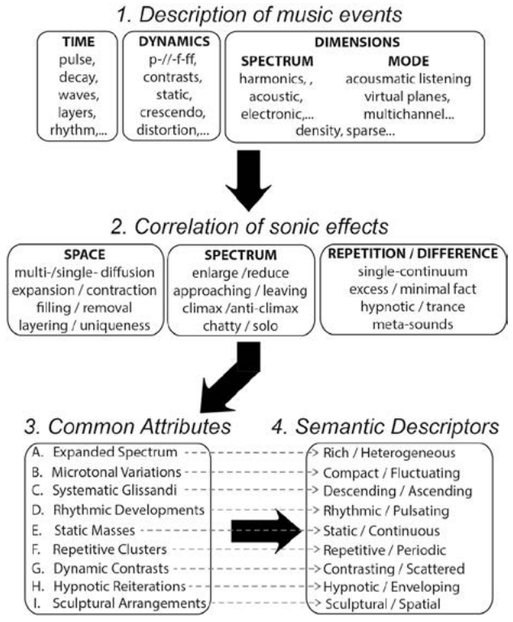


Figure 1. The analysis led to the identification of nine musical attributes (step 3), these were translated into corresponding semantic descriptors (step 4) to be used in Perceptual Studies.

3.4 Some Considerations

In some pieces, these attributes are frequently combined, *e.g.* glitch-electronic music usually exhibits repetitive clusters (F) within rhythmic frameworks (D), while the repetition of musical units (F) in Lang and Haas' pieces can at times be associated with non-rhythmic hypnotic reiterations (H) or more complex structures.

More generally, it can be seen that contemporary instrumental compositions make elaborate use of simple musical elements to create new effects: the reiteration of descending notes at the beginning of *in vain* evokes a sort of a cyclic falling whirl that constantly ascends. Similarly, electronic pieces apply drastic timbric techniques to arrive at analogous results, such as the application of periodically fluctuating frequency filters in the music of Pan Sonic (*Rafter*). When more static musical episodes are considered or even when, in the case of Lang's or Nova's pieces, electronic devices are used in notated compositions there is an explicit intention to take inspiration from wider cultural aesthetics, such as turntablism, glitch or noise for Lang, Nova and Verrando, respectively.

One might argue that this selection of pieces includes strategically chosen examples to facilitate the method of analysis and that this analysis occurs only at a relatively general sonic level. However, as this cross-genre examination is innovative and untested, it was important to start with a solid and fertile set of musical works in order to create a clear template to be developed in the future.

Considering the overall results of this analysis, each work displays at least eight out of the nine attributes identified above. Therefore, even though these nine attributes are fairly general, their presence across all pieces represents a starting point for the definition of a common cross-genre perspective.

Now, in order to confirm if these evidences are essential, it is necessary to move from analytical approach to empirical studies. These attributes (Section 3.3) represent, in fact, the framework for the extension of the study toward perceptual examinations.

4. PERCEPTUAL STUDIES

The question of the perceptual aspect of sound spans an immense area of studies from psychology [12] to neurosciences [13] and concerns human reactions that extend over stimuli, feelings and emotions [14]. The majority of works in music perception makes use of several strategies (*e.g.* semantic differential, multidimensional scaling, verbal attribute magnitude estimation…) moving between two approaches: on the one hand, some analytical assessments aim to explore sound's qualities and timbre, using adjectives of different semantic classes to define simple sounds that are often uniform (*i.e.* single tones or noises) in order to have a consistent response in their examinations. On the other hand, some studies have to do more with philosophy, semiology and psychology and focuses on formalist, cognitivist and emotivist positions and refer to western classical repertoire and 20th century popular music. Within the large production of studies in music perception there is still a reduced number of works that deals with contemporary experimental music.

The lack of perceptual studies in experimental music is a sort of paradox, because many genres of exploratory music, today, deal with perceptual aspects more than ever. The interest in perception of new music seems to have moved from scholars towards composers themselves. When these two figures coincide we find interesting debates: for instance, the intense literary production within the electroacoustic communities [10] is an example of how profound is the interest in listening, perception and cognition of music within some communities.

Coming to this study, how should be treated, then, a selection of pieces with no tonal construction, no traditional narrative and time perception but containing noises, real-world elements, acoustic, electronic and manipulated sounds? This task aims to (i) evaluate the similarities, previously identified in theoretical analysis; (ii) investigate the capacity of listener to express about his perception of sound; (iii) verify if one or more styles of music are actually considered in a different manner; (iv) correlate the answer with the typology of listener: his musical training, preferences and background.

The listener's response to abstract and experimental works of music relates to rational and conceptual issues involving listener's background, but also with phenomenal qualities and pure sonic stimuli

On the one hand, the musical works selected for this study contain elaborated constructions, and consequently I could not use simple descriptors, such as "the sound is sharp or dull", to describe, for instance, a composition for 24 instruments. These pieces include complex sounds and structured textures thus being at the same time *sharp* and *dull*. On the other hand, a more detailed comparison of the emotional effect of music would not furnish useful results; the emotional side is differently conveyed in case of a contemporary composition or an electronic club-based session: there are different purposes, enjoyments, stimuli and interests. Therefore, the emotive responses to these pieces could be greatly different and are not the central part of this investigation. Rather, I would examine the ability of different typologies of listeners to express and distinguish sonic characteristics.

For this reason, I arranged an array of descriptors starting from the nine musical attributes identified in the analytic step. This verbal translation (Figure 1, steps 3 and 4) into more comprehensible adjectives would favour the extension of the study to untrained listeners that may be unfamiliar with the meaning of specific words or idiomatic expression. However, this linguistic conversion is strictly related with the musical characteristics of the pieces and is not a universal transformation adaptable for every type of music. For instance, exclusively for this study, it has been possible to adapt the expression "microtonal variation" into the couple of adjectives "compact / fluctuating". Within the selection of pieces, in fact, the use of microtonality is encountered in sustained musical episodes and concerns with the creation of static blocks of sound (*e.g. compact*) with aural fluctuations (*e.g. fluctuating*) generated by acoustic binaural beats.

To summarize, considering a double approach, where, on the one hand, there are practices typical of a timbre assessment, through the use of pure tones and noises, and on the other hand we find surveys that explore the emotional response towards a traditional repertoire; I may virtually place this study midway between these two approaches.

4.1 Method

The examination consisted on a listening session (nine excerpts –approx. 1 min. each– from the selection of pieces, see Section 3.2) combined with an evaluation questionnaire and it has been carried out both through a web-based platform and some direct experimental surveys. The participants (N=55) were mainly students and young musicians in the age of 20 to 40 years.

The first section includes four questions to define the typology of participants (*i.e.* age, professional link to music, time of listening music in everyday life, music preferences). After that, participants could listen at the selection of the audio samples and they should answer about familiarity with the types of music samples. In the next section (Perceptual Evaluation #1), participants are invited to sort the audio samples into groups and to indicate which criteria they apply (it is specified that any criteria is acceptable and it is not compulsory to separate the samples). Subsequently (Perceptual Evaluation #2), the list of semantic descriptors (Figure 1, step 4) is provided to the participants, who are asked to associate these descriptors to audio excerpts.

4.2 General remarks on the perceptual survey

There is a prevalence of trained listeners (N=38, 68%): several personal contacts with potential participants confirmed that when untrained listeners approach the questionnaire they often left it when started to listen at the selection of pieces. They stated to feel "incompetent and unsuitable for the type of music" and become reluctant to participate.

Nevertheless the variety of typologies and musical preferences allows to traces some preliminary notations:

• there is mutual correspondence among musical preferences, familiarity with the audio samples and the questionnaire evaluation, thus indicating that besides this work extend across various genres of contemporary music is still contemplated as a "niche", limited and isolated branch of today's music.

• Trained participants tend to group sample based on instrumentation and recognized genres, while untrained participants also use personal sensations and feeling as parameters to classify. The results seem to suggest the most logic outcome: most of the participants (62%) is inclined to categorize pieces based on their past experience than based on their transitory sensations.

• There is a gap between trained and untrained participants. Even if I have attempted to modulate the questions in order to be understandable to all listeners, those with experiences in abstract and experimental music show a greater ability (i) to distinguish styles and genres; (ii) to identify the nature and the source of different sounds; (iii) to deal with semantic descriptors of different spheres of sense other than hearing (*e.g.* visual or tactile). In particular, when the music shows several attributes, trained participants are able to better identify them than untrained ones; whereas the piece exhibit few specific attributes, the experienced listeners succeed to improve his answer indicating additional minor attributes that the inexperienced listeners are not capable to recognize.

• Looking at the Perceptual Evaluation #1 (sorting task), there is a positive and interesting response: there is a significant number of participant (80%) able to group all the samples, accomplish multiple connections and classifications and provide detailed descriptions of the criteria they used. This evaluation (PE1) tells us that the major sorting criterion appears to be the recognized instrumentation and style. At the second stage, participants consider the atmosphere and character of a piece. The results confirm essentially that a common platform among distant genres could exist if this second criterion that concerns more with musical atmosphere and sonic character and effect, is not just a secondary aspect but holds a important role. A future study should (i) investigate longer audio samples, (ii) focuses on the aural effect of the music and (iii) account for a more profound depiction of piece's sonic nature.

• The Perceptual Evaluation #2 has been accomplished more successfully than PE1: it seems to be more demanding to define own musical decisions and express them with own words (PE1) than combine and associate given expressions to what one had listened before (PE2). In general, untrained listeners handle better generic descriptors and those that relate more with effect of music (*e.g.* "hypnotic", Figure 1, step 4) than with intrinsic qualities. On the contrary, trained listeners prefer functional and structural descriptors (*e.g.* "sculptural", "fluctuating"). In the future, it could be important to differentiate these classes and explicitly declare when it is aimed to inspect inner quality of sounds and their effect.

5. CONCLUSIONS

This study focus on a specific facet of today's experimental music, selecting works that favour the exploration of sound itself over more structured writing techniques or systems. On the one hand, this study looked at styles of music that deal with a limited set of characteristics (*e.g.* hypnotic effects, repetition, rhythm) but which make use of innovative sonic contrasts and complex elaborations of sound. On the other hand, compositions belonging to the so-called contemporary instrumental music genre were seen to hold similar qualities within more elaborate structures.

This paper does not intend to over- or underestimate any particular genre of music, but aims rather to highlight interesting correspondences between distant genres according to specific uses of sonic material and so originate constructive debates. It strives to compare (other than analyse) different styles based on a more aesthetic approach, trying to express it through descriptive attributes able to define more clearly this cross-genres perspective. The latter, in fact, even if it could be likely well know among scholars and trained listeners, continues to be indistinct and not fully recognized.

Perceptual studies reveal that there is still a virtual barrier that separates the world of exploratory music (academic and independent) and the study highlight the difficulties that inexperienced listeners encounter approaching a diverse material of experimental music. However, this work suggest that it would be possible to set a series of informative tools for young students and listeners to approach this type of music. Moreover, the study helps to define the potential of a semantic descriptor when connected to the field of contemporary exploratory music (*e.g.* how an adjective works, where and how it should be applied…).

Finally, the paper (i) shows an advanced strategy that combines perceptual studies with theoretical analyses to define a more profound cross-genres perspective over experimental fields of today's music; (ii) it displays correlations among these audio extracts and open the way to reflect in how distant pieces could be treated; (iii) it permits a better understanding of various fields of music and would facilitate artistic convergences and (iv) it would help in the creation of didactic and academic platforms for the study of diverse musical contexts within a unified framework.

Acknowledgments

The author would thank Fundação para a Ciência e a Tecnologia (FCT) of Portugal for a doctoral fellowship (SFRH/BD/102506/2014) and all participants from Lisbon, Paris and Huddersfield who collaborated in this research.

6. REFERENCES

[1] C. Cox and D. Warner, *Audio Culture. Readings in Modern Music*. Bloomsbury, 2007.

[2] P. Griffiths. *Modern Music And After*. Oxford University Press, 2010.

[3] M. Solomos, *De La Musique Au Son.* Presse Universitaires de Rennes, 2013.

[4] J. Demers, *Listening Through Noise.* Oxford University Press, 2010.

[5] R. Weale, "The Intention/Reception Project", *PhD Thesis*. De Montfort University, 2005.

[6] R. Wanke, "A Cross-genre Study of the (Ec)Static Perspective of Today's Music", *Organised Sound*, vol. 20, pp. 331–339, 2015.

[7] S. Voegelin, *Listening to Noise and Silence.* Continuum Books, 2010.

[8] S. Roy, *L'analyse Des Musiques Électroacoustiques.* Harmattan, 2003.

[9] D. Smalley, "Spectromorphology: explaining sound-shapes", *Organised Sound*, vol. 2, pp. 107-126, 1997.

[10] S. Emmerson and L. Landy. *Expanding the Horizon of Electroacoustic Music Analysis.* Cambridge University Press, 2016.

[11] K. Cascone, "The Aesthetics of Failure: 'Post-Digital' Tendencies in Contemporary Computer Music", *Computer Music Journal,* vol. 24, no. 4, pp. 12–18, 2000.

[12] S. McAdams, in *Thinking in Sound: The Cognitive Psychology of Human Audition*, (eds) S. McAdams and E. Bigand, 146–198, Oxford Univ. Press, 1993.

[13] D. Pressnitzer, A. de Cheveigné, S. McAdams and L. Collet (Eds.), *Auditory Signal Processing*. Springer-Verlag, 2005.

[14] P. N. Juslin, "From everyday emotions to aesthetic emotions: Towards a unified theory of musical emotions", *Physics of Life Reviews*, vol. 10, no. 3, pp. 235–266, 2013.

Diegetic Affordances and Affect in Electronic Music

Anıl Çamcı
University of Illinois at Chicago
anilcamci@gmail.com

Vincent Meelberg
Radboud University Nijmegen
v.meelberg@let.ru.nl

ABSTRACT

In this paper, we investigate the role affect plays in electronic music listening. By referring to a listening experiment conducted over the course of three years, we explore the relation between affect and diegetic affordances (i.e. those of the spatiotemporal universes created by electronic music). We will compare existing perspectives on affect with the psychologist James Gibson's model of affordances in the context of an electronic music practice. We will conclude that both the sounds themselves and the diegetic affordances of these sounds may elicit affective reactions, and that further study into the relation between diegesis, affordance, and affect may contribute to a better understanding of what we hear in electronic music.

1. INTRODUCTION

In contemporary music studies, *affect* seems to play an increasingly important role. This concept enables the articulation of the way music has an impact on listeners and artists alike. In this paper, we explore how affect "works" in electronic music, and how it is intrinsically related to affordances of an electronic music experience. More specifically, we will discuss the manners in which so-called diegetic affordances may evoke affects with listeners.

Listeners of electronic music may derive diegeses (i.e. spatiotemporal universes referred to by narratives) from the poietic trace left by the composer. In semantic consistency with these diegeses, listeners populate the landscapes of their imaginations with appropriate objects, situated in various configurations based on cognitive or perceptual cues. As they do so, they also experience this environment with implied affordances true to the objects of their imagination, and affects attached to these diegetic possibilities.

First, we will outline a listening experiment, the results of which will be used to further our discussion. We will then define affects and affordances, and how these can be applied in the articulation of musical experience. We will explore the similarities between these two concept to construct a framework that can be used to articulate artistic experiences. In doing so, we will suggest that the diegetic affordances of electronic music may evoke affective responses with listeners. Finally, based on the listener feedback gained from the experiment, we will demonstrate how this framework can be useful when discussing features of electronic music that are corporeally relevant for the listener.

2. OVERVIEW OF THE EXPERIMENT

Between May 2012 and July 2014, 60 participants from 13 different nationalities took part in a listening experiment that investigates the cognition of electronic music. 23 participants were female while 37 were male. The average age of the participants was 28.78. Ages ranged from 21 to 61. 22 participants identified themselves as having no musical background. Amongst the remaining 38 participants were musicians, music hobbyists, composers, and students of sound engineering and sonic arts.

The experiment aimed to explore how fixed works of electronic music operate on perceptual, cognitive and affective levels. The design of the experiment was aimed at extracting both contextual and in-the-moment impressions while offering a natural listening experience. The design involved: 1) an initial listening section, where the participants were asked to listen to a complete work of electronic music without any instructions pertaining to the experiment, 2) a general-impressions task, where the participants were allowed to reflect upon their experience in writing without any form or time constraints, 3) a real-time input exercise, where the participants were acquainted with a browser-based system in which they could submit descriptors in real time while hearing an audio material, and 4) a real-time free association task, where the participants listened to the same piece they heard earlier while at the same time submitting descriptors as to anything they might feel, imagine or think as they listen to the piece.

Five complete pieces of electronic music, in 44.1 kHz, 16-bit WAV format, were used in the experiments. Four of these pieces, namely *Birdfish*, *Element Yon*, *Christmas 2013*, and *Digese*, were composed by the first author of this paper. The fifth piece was Curtis Roads' 2009 piece *Touche pas*. Said pieces utilize a wide range of forms, techniques (e.g. live performance, micromontaging, algorithmic generation), tools (e.g. audio programming environments, DAWs, physical instruments) and material (i.e. synthesized and recorded sounds).

The results of this experiment, including a categorical analysis of the real-time descriptors and a discourse analysis of the general impressions, have been offered in previous literature [1, 2]. The current paper relies primarily on a semantic analysis of the general impressions and the real time descriptors. The general-impressions were provided in one or a combination of various forms, including list of words, list of sentences, prose and drawings. The vast majority of the descriptors submitted in the real-time free association task were single words or two-word noun phrases. A participant's prior experience with electronic music did not significantly impact the semantic qualities (e.g. representationality versus abstractness) or the number of the descriptors submitted by that participant. Technological listening, where a listener recognizes the technique behind a work [3], was infrequently apparent in the responses by sonic arts students.

3. AFFECT IN MUSIC

3.1 Interpretations of Affect

The affective appraisal of music comprises successive stages that utilize different but interconnected perceptual resources. A particular component of this spectrum is the experience of affect, which has been studied within a variety of domains ranging from virtual reality [4] and painting [5] to politics [6] and sports [7]. This concept is not only adopted by a large array of disciplines but also subjected to a variety of interpretations. On the far end of the spectrum, Lim et al. [8] and Shouse [9] point to uses of affect as a synonym for emotion. While this approach begs the question of why affect would need to be demarcated as a separate concept, it nevertheless provides an insight regarding the context within which the concept is situated.

The use of affect in philosophy dates back to Spinoza's *Ethics*. Spinoza identifies affect as an affection of the body by which "the body's power of acting is increased or diminished" [10]. In his introduction to Deleuze and Guattari's *A Thousand Plateaus*, the philosopher Brian Massumi offers a related description of affect as a "prepersonal intensity corresponding to the passage from one experiential state of the body to another" [11]. Emotion on the other hand is personal according to Massumi: "Emotion is qualified intensity, the conventional, consensual point of insertion of intensity into semantically and semiotically formed progressions, into narrativizable action-reaction circuits, into function and meaning" [12].

Based on Massumi's interpretation, we have previously proposed the concept of a sonic stroke [13]. A sonic stroke is an acoustic phenomenon that induces musical affect upon impacting the listener's body. A consequence of this impact is emotion, which emerges once the affect is reflected upon (i.e. a sonic stroke is registered as a musical gesture).

3.2 Affect and Mechanisms of Music Perception

Music, despite lacking immediate survival value, activates brain mechanisms associated with pleasure and reward. The combined sensory and cognitive experience of a musical piece influences the listener's affective state [14]. Accordingly, existing research points to a mixture of cultural and physiological determinants of music appreciation [15, 16]. Brown et al. delineate musical universals, such as loudness, acceleration and high-registered sound patterns, which incite affective experience independent of cultural origin [17].

Juslin and Västfjäll emphasize a need to investigate the mechanisms underlying the affective appraisal of music [18]. They argue that the evocation of emotions in music is based on processes that are not exclusive to music. They enumerate several neural mechanisms that contribute to this phenomenon. Out of these, the *brain stem reflex* deals with the low-level structural and cross-cultural characteristics of the musical experience. Brain stem reflexes are hard-wired and are connected with the early stages of auditory processing. Sounds that are sudden, loud, dissonant, or those that feature fast temporal patterns signal the brain stem about potentially important events and induce arousal. This arousal reflects the impact of auditory sensations in the form of "music as sound in the most basic sense" [18].

Due to its attachment to the early stages of auditory processing, brain stem reflex is highly correlated with human physiology and the so-called universals (i.e. the low-level structural properties) of musical experience. A functional coherence between affect and the brain stem reflex is highlighted by their intrinsic reliance on the spectrotemporal and dynamic properties of musical sound. While affect represents the corporeal segment of the affective appraisal of music, it cannot be dissociated from an ensuing emotion. This is mainly due to the aforementioned interplay between the mechanisms underlying music cognition. The musicologist Marc Leman points to seminal neuroscientific studies, such as those by Antonio Damasio, Marc Jeannerod and Wolf Singer, that motivate a departure from the Cartesian view of "mind and matter" as separate entities; it is understood that the so-called subjective world of mental representations stems from our embodied interactions with the physical environment [19].

4. AFFORDANCES

An approach to perception that is commonly facilitated in musical research [20, 21, 22] is the model of affordances developed by the psychologist James Gibson. Gibson's studies on ecological perception stemmed from his experiments in aviation during the World War II. Focusing mainly on an active observer's perception of its environment, Gibson postulated that the invariant features of visual space represent pivotal information for perception. *I*nvariants are features of an object that persist as the point of observation changes [23]. While most items in Gibson's taxonomy of invariants pertain to the visual domain, his concept of affordances has been applied to other modalities of perception including hearing.

According to Gibson, objects in an environment, by virtue of their invariant features, afford action possibilities relative to the perceiving organism. For instance, a terrestrial surface, given that it is flat, rigid and sufficiently extended, affords for a human-being the possibility to walk on it [23]. His main motivation to propose this seemingly straightforward idea is to refute the prevailing models of perception, which assume that ecological stimuli are chaotic, and therefore the perceiver must extract a meaning out of sensory stimuli by imposing mental structures upon disorganized information. Gibson suggests that there are certain kinds of structured information available prior to perception in the form of invariants. The nature of these invariants is relative to the complexity of the perceiving animal [24]. In other words, an object will have different affordances

for different perceivers: a stone, on account of its physical characteristics, affords the action possibility of throwing for a human-being, while at the same time affording the action possibility of climbing for an ant.

Gibson suggests that "perceptual seeing is an awareness of persisting structure" [23] and that knowledge exists in the environment, for a viewer to pick up. When viewed in the light of modern experimental studies on perception, we can consider Gibson's proposal of perceptual knowing as an addition to, rather than a replacement for, the existing models of learning that are based on memory processes. This sentiment is clearly materialized in Gibson's writing as well when he states: "To perceive the environment and to conceive it are different in degree but not in kind. One is continuous with the other" [23]. The ecological approach addresses certain stages of our perceptual experience and complements higher-level mental processes. In that respect, Gibson's model of invariants aligns with various models of experience, such as perceptual symbols and schemas [25].

5. DIEGETIC AFFORDANCES AND AFFECT

As the above discussion indicates, the concept of affect and the model of affordances have significantly convergent characteristics, even though they emerge from two separate fields of study, namely philosophy and psychology. Recalling our previous discussions of these concepts, a correspondence chart between these two concepts, as seen in Table 1, can be formed.

Affordance	**Affect**
Pre-personal, structured information available in the (material) environment	Pre-personal intensity
Precedes cognitive processes	Unqualified experience
Action possibility	Affective potentiality
Relative to the observer's form	A corporeal phenomenon

Table 1. A comparison of the definitions of affordance and affect

This table shows that how these two concepts, by their definitions, are contiguous with each other. Both represent capacities, one pertaining to the perceived object and the other to the perceiver. If a link is therefore to be formed between the two, an affordance can be characterized as inductive of affect. While Massumi characterizes emotion as a sociolinguistic fixing of the experiential quality that is affect, he later underplays the one-way succession of affects into emotions by stating that affect also includes social elements, and that higher mental functions "are fed back into the realm of intensity and recursive causality" [12]. Affects, anchored in physical reality, are therefore both pre- and post-personal. This dual take on affect is also apparent in Freud's interpretation of the concept: unconscious affects persist in immediate adjacency to conscious thoughts and they are practically inseparable from cognition [26].

In their article *Percept, Affect, and Concept*, Deleuze and Guattari elegantly describe "how the plane of the material ascends irresistibly and invades the plane of composition of the sensations themselves to the point of being part of them or indiscernible from them" [27]. Affect, as we would like to therefore interpret it, represents a landscape of experiences from which emotions sprout. This landscape is superimposed on the material. The affordances of the material evokes affects with the perceiver. An object represented in electronic sound constitutes a material of second order which induces an affective experience. Simultaneously with the ascension of the embodied sound into affect, the representation ignites an affective thread of its own. The imagined spatiotemporal universe evoked by this representation will have its own dimensions, landscapes, surfaces and objects.

However, such landscapes and surfaces will only afford so-called diegetic action possibilities to the listener. The narratologist Gérard Genette defines diegesis as "the spatiotemporal universe" referred to by a narrative [28]. The concept of diegesis can be traced back to Plato's dichotomization of narrative modes into imitation and narration [29]. However, it has since yielded various incarnations that have been used for describing narrative structures in art, and situating the components of an artwork in relation to one another. On a meta-level, the resulting narratological perspectives also provide insights into the fabric of the artistic experience by delineating relationships between the artist, the artistic material and the audience [30]. Differentiating between cascading layers of a narrative by starting from the physical world of the author on the outermost level, Genette outlines the concept of diegesis as the spatiotemporal universe to which the narration refers. Therefore, in his terminology, a diegetic element is "what relates, or belongs, to the story" (translated in [31]). And it is precisely such an imagined spatiotemporal universe, in this case created as a result of listening to electronic music, that we describe as consisting of diegetic affordances that are evocative of affects [30].

Gibson describes a behavior for surrogate objects in the visual domain, such as a photograph or a motion picture, that is similar to the diegetic action possibilities introduced above [23]. While these objects also specify invariants, they instigate indirect awareness and provide "information about" [24]. The electronic music listener can also make out acoustic invariants characteristic of a certain object. While a representation in electronic music will be "a structured object in its own right" [22], the action possibility will nevertheless remain virtual for the listener since the imagined object is an external representation:"[t]he perception or imagination is vicarious, an awareness at second hand" [23].

Affects are semantically processed, fed back into the established context and experienced as the result of diegetic affordances. When watching a horror movie for instance, the viewers are aware that they are in a theatre. But once they have been acculturated into the story of the film, a mundane and seemingly non-affective act, such as switching on the lights in a room, becomes loaded with affect, because threat, as an affect, "has an impending reality in the present" [6]. Listeners of electronic music concoct diegeses from the poietic trace left by the composer. In semantic consistency with these diegeses, listeners populate the landscapes of their imaginations with appropriate objects, situated in various configurations based on cognitive or perceptual cues. As they do so, they also experience this environment with implied affordances true to the objects of their imagination, and affects attached to these diegetic possibilities. As Gibson explains:

> The beholder [of a film] gets perception, knowledge, imagination, and pleasure at second hand. He even gets rewarded and punished at second hand. A very intense empathy is aroused in the film viewer, an awareness of being in the place and situation depicted. But this awareness is dual. The beholder is helpless to intervene. He can find out nothing for himself. He feels himself moving around and looking around in a certain fashion, attending now to this and now to that, but at the will of the film maker. He has visual kinesthesis and visual self-awareness, but it is passive, not active. [23]

Accordingly, the listener of electronic music experiences passive aural kinesthesis. An inexperienced participant, who listened to *Digese*, narrated a highly visual story of her experience in her general impressions:

> Glass/metal ping pong balls are constantly being dropped on the floor as we walk through an empty salon with bare feet; we leave this room and go out in a jungle, moving through the grass stealthily; passing through cascading rooms; we arrive in another salon.

While many of the objects in her narrative also appear in descriptors provided by other participants, details like "walking with bare feet" and "moving stealthily" are indicative of the participant's individual affective experience of the diegetic environments of her imagination.

6. THE AFFORDANCES OF IMAGINED SOUND SOURCES IN ELECTRONIC MUSIC

The concept of diegetic affordances can be useful when discussing features of electronic music that are corporeally relevant for the listener. Reverberation, for instance, affords a relative sense of space while low frequency gestures afford an awareness of large entities. Even purely synthesized sounds can afford the instigation of a mental association to a sound source. When a source for a sound object is imagined, the mind will bridge the gaps as necessary to achieve a base level of consistency by attributing featural qualities to the source. In literature, this is referred to as the "principle of minimal departure", which describes the readers' tendency to relate a story to their everyday lives in order to resolve inconsistencies or fill the holes in the story. In electronic music, this tendency is informed by our mental catalogue of auditory events we have been exposed to thus far: we possess a sophisticated understanding of how a certain object in action will sound in a certain environment. As the design researcher William Gaver states, the material, the size and the shape of a physical object will intrinsically determine how the object vibrates and therefore produces sounds: for instance, vibrations in wood damp much more quickly than in metal, "which is why wood "thunks" and metal "rings" and big objects tend to make lower sounds than small ones" [32].

Therefore, even the most elementary attributes of a sound can indicate a physical causality. In that respect, granular synthesis bears a significant capacity. In granular synthesis, the metaphorical relationship between a microsound and a particle can be extended to a physical model. In the experiment results, gestures produced using granular synthesis were described by various participants as particles (pieces, cells, glass, metal) dividing (breaking) and merging (coming back together, colliding). These reports highlight the implication of a mechanical causality inherent to granular synthesis. The frequency and the amplitude envelope of a grain can be altered to specify a particle's size. *Touche pas* is particularly rich in similarly shaped objects of various sizes, as evidenced in the real-time descriptors referring to spherical objects of diverse proportions. Furthermore, the timbral characteristics of grains can be altered in order to imply different surface materials. In *Digese*, which quotes a particular granular texture from *Touche pas*, listeners differentiated between timbral varieties by defining different material types and objects. Separate participants described imagining "glass/metal balls", "ping pong balls", a "pinball machine", "champaign" (cork sound), a "woodpecker" and "knocking on the door". Here the materials vary from metal and plastic to wood. For *Touche pas* both "coins", "marbles", "ping pong balls", "bowling ball" and "xylophone" were submitted as descriptors, indicating a similar spectrum of materials.

An important determinant of such descriptors is the motion trajectory of a grain. The particular motion trajectory used in *Digese* is inspired by the concurrent loops of unequal durations heard in Subotnick's seminal piece *Touch*, a behavior which is also apparent in *Touche pas*. When multiple loops are blended together, the resulting texture implied for most participants a sense of "bouncing" (i.e. "marbles bouncing") or "falling" (i.e. "rocks falling together"). One participant wrote: "the clicking sounds (...) resembled a dropped ball bouncing on a surface, since each sound came in slightly quicker than the previous one". Another participant described *Touche pas* as displaying a "convincing physicality". Once a motion trajectory is coupled with the imagined material of the object, higher-level semantic associations occur: while one participant described "bouncing on wood" followed by "marimba", another participant wrote that marbles made her think about "childhood", "fun" and "games".

The cognition of motion trajectories can be a function of temporal causality. The researcher Nancy VanDerveer draws attention to temporal coherence as "possibly the primary basis for the organization of the auditory world into perceptually distinct events" [33]. To examine the effects of temporal factors in the identification of environmental sounds, Gygi et al. used event-modulated noises (EMN) which exhibited extremely limited spectral information [34]. By vocoding an environmental sound recording with a band-limited noise signal, the event-related information was reduced to temporal fluctuations in dynamics of a spectrally static signal. From experiments conducted with EMN, re-

searchers concluded that, in the absence of frequency information, temporal cues can be sufficient to identify environmental sounds with at least 50% accuracy. Articulation of a so-called physical causality through the temporal configuration of sound elements is apparent in most of the pieces we used in our experiment, and particularly in gestures that bridge consecutive sections of a piece (e.g. 0‘33” to 0‘39” in *Christmas 2013* and 1‘27” to 1‘30” in *Digese*).

In *Birdfish*, short-tailed reverberation and low frequency rumbles were utilized to establish the sense of a large but enclosed environment. These were reflected in the real-time descriptors with such entries as “cave”, “dungeon” and “big spaceship”. Similar cues in *Christmas 2013* prompted listeners to submit “open sea”, “open space” and “sky” as descriptors. The spectral and reverberant attributes of the sound specify environments in various spatial proportions with the listener. This information implies, for instance, the affordance of locomotion (which in several cases manifested itself as that of “flying”).

In *Element Yon*, which inhabits a strictly abstract sound world, the frequency and damping characteristics of certain gestures instigated such descriptors as “metal balls getting bigger and smaller”, “high tone falls and hits the ground”. Here, distinctly perceptual qualities are situated in metaphors, while retaining their embodied relationship with the listener. Another similar example is observed in the responses to gestures with high frequency content in *Birdfish*, which listeners characterized with such descriptors as “ice”, “glass”, “metal”, “blade” and “knife”. These descriptors imply both a metaphorical association and an affordance structure between high frequencies and a perceived sense of sharpness.

Many descriptors submitted by the participants of the experiment denoted living creatures. However, a portion of these source descriptors were augmented by featural descriptors to form such noun phrases as “tiny organisms”, “baby bird”, “little furry animal”, “huge ant” and “huge animal”. Here, featural descriptors signify the proportions of the perceived organisms. In these cases, featural information available in the sounds afforded the listeners a spatial hierarchy between the imagined creatures and themselves.

The linguist John Ohala points to the cross-species association of high pitch vocalizations with small creatures, and low pitch vocalizations with large ones [35]. He further delineates that the size of an animal, as implied by the fundamental frequency of its vocalizations, is also an indicator of its threatening intent. Based on Ohala’s deductions, the spatial extent of an organism communicated in its vocalization characteristics, which would possess a survival value in a natural environment, is an affordance of threat. Featural descriptors can therefore be viewed as indicative of affect.

Gliding pitch variations in intonation are expressive of not only meaning [36] but also personality and emotion [37]. Furthermore, this is true not only of humans but also of vocalizing animals in general [38]. The gestures consisting of rapid frequency modulations of monophonic lines in *Element Yon* were therefore suggestive of an organic origin, as evidenced in descriptors such as “I guess he is trying to tell us something”, “communication”, “conversation”, “crying”, “scream”.

7. CONCLUSIONS

Music listening is a complex activity in which affect plays a crucial role. As our discussion of listening to electronic music has revealed, both the sounds themselves and the diegetic affordances of these sounds may elicit affective reactions. It is the latter kind of affective reaction, in particular, that has a decisive influence on the manner in which electronic music may be interpreted by listeners. In the experiment results, we observed that diegetic affordances guide the listeners to higher-level semantic associations, which inherently inform their affective interpretation of a piece. As a consequence, we believe that further study into the relation between diegesis, affordance, and affect may contribute to a better understanding of what we hear in electronic music.

8. REFERENCES

[1] A. Çamcı, “A cognitive approach to electronic music: theoretical and experiment-based perspectives,” in *Proceedings of the International Computer Music Conference*, 2012, pp. 1–4.

[2] ——, “The cognitive continuum of electronic music,” Ph.D. dissertation, Academy of Creative and Performing Arts (ACPA), Faculty of Humanities, Leiden University, Leiden, 2014.

[3] D. Smalley, “Spectromorphology: explaining sound-shapes,” *Organised sound*, vol. 2, no. 02, pp. 107–126, 1997.

[4] L. Bertelsen and A. Murphie, “Félix Guattari on Affect and the Refrain,” in *The affect theory reader*, M. Gregg and G. J. Sigworth, Eds. Durham: Duke University Press, 2010, p. 138.

[5] G. Deleuze and F. Bacon, *Francis Bacon: the logic of sensation*. Minneapolis: University of Minnesota Press, 2003.

[6] B. Massumi, “The political ontology of threat,” in *The affect theory reader*, M. Gregg and G. J. Sigworth, Eds. Durham: Duke University Press, 2010, pp. 52–70.

[7] P. Ekkekakis, *The measurement of affect, mood, and emotion: a guide for health-behavioral research*. Cambridge: Cambridge University Press, 2013.

[8] Y.-k. Lim, J. Donaldson, H. Jung, B. Kunz, D. Royer, S. Ramalingam, S. Thirumaran, and E. Stolterman, “Emotional experience and interaction design,” in *Affect and Emotion in Human-Computer Interaction*, C. Peter and R. Beale, Eds. Berlin: Springer, 2008, pp. 116–129.

[9] E. Shouse, “Feeling, emotion, affect,” *M/C Journal*, vol. 8, no. 6, p. 26, 2005.

[10] B. de Spinoza, *A Spinoza reader*, E. Curley, Ed. Princeton: Princeton University Press, 1994.

[11] G. Deleuze and F. Guattari, *A thousand plateaux*. Minneapolis: University of Minnesota Press, 1987.

[12] B. Massumi, *Parables for the virtual: movement, affect, sensation*. Durham: Duke University Press, 2002.

[13] V. Meelberg, “Sonic strokes and musical gestures: the difference between musical affect and musical emotion,” in *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009)*, 2009, pp. 324–327.

[14] V. N. Salimpoor, I. van den Bosch, N. Kovacevic, A. R. McIntosh, A. Dagher, and R. J. Zatorre, “Interactions between the nucleus accumbens and auditory cortices predict music reward value,” *Science*, vol. 340, no. 6129, pp. 216–219, 2013.

[15] S. E. Trehub, “Human processing predispositions and musical universals,” in *The origins of music*, N. L. Wallin, B. Merker, and S. Brown, Eds., 2000, pp. 427–448.

[16] M. E. Curtis and J. J. Bharucha, “The minor third communicates sadness in speech, mirroring its use in music.” *Emotion*, vol. 10, no. 3, p. 335, 2010.

[17] S. Brown, B. Merker, and N. L. Wallin, *An introduction to evolutionary musicology*. Cambridge: MIT Press, 2000.

[18] P. N. Juslin and D. Västfjäll, “Emotional responses to music: the need to consider underlying mechanisms,” *Behavioral and brain sciences*, vol. 31, no. 05, pp. 559–575, 2008.

[19] M. Leman, *Embodied music cognition and mediation technology*. Cambridge: MIT Press, 2008.

[20] S. Östersjö, “Shut up ’n’play,” Ph.D. dissertation, Malmö Academy of Music, Malmö, 2008.

[21] W. L. Windsor, “A perceptual approach to the description and analysis of acousmatic music,” Ph.D. dissertation, City University, London, 1995.

[22] C. O. Nussbaum, *The musical representation: Meaning, ontology, and emotion*. Cambridge: MIT Press, 2007.

[23] J. Gibson, *The ecological approach to visual perception*, ser. Resources for ecological psychology. Mahwah: Lawrence Erlbaum Associates, 1986.

[24] J. J. Gibson, *The senses considered as perceptual systems*. Boston: Houghton Mifflin, 1966.

[25] D. A. Schwartz, M. Weaver, and S. Kaplan, “A little mechanism can go a long way,” *Behavioral and Brain Sciences*, vol. 22, no. 04, pp. 631–632, 1999.

[26] M. Gregg and G. J. Seigworth, *The affect theory reader*. Durham: Duke University Press, 2010.

[27] G. Deleuze and F. Guattari, *What is philosophy?* New York: Columbia university Press, 1994.

[28] G. Genette, *Figures I*. Paris: Seuil, 1969.

[29] Plato, *Plato, The Republic*, S. H. D. P. Lee, Ed. London: Penguin Books, 1955.

[30] A. Çamcı, “Diegesis as a semantic paradigm for electronic music,” *eContact*, vol. 15, no. 2, 2013, [Online], http://econtact.ca/15_2/camci_diegesis.html. Accessed May 12, 2016.

[31] R. Bunia, “Diegesis and representation: beyond the fictional world, on the margins of story and narrative,” *Poetics Today*, vol. 31, no. 4, pp. 679–720, 2010.

[32] W. W. Gaver, “What in the world do we hear?: an ecological approach to auditory event perception,” *Ecological psychology*, vol. 5, no. 1, pp. 1–29, 1993.

[33] N. J. VanDerveer, “Ecological acoustics: human perception of environmental sounds.” Ph.D. dissertation, ProQuest Information & Learning, 1980.

[34] B. Gygi, G. R. Kidd, and C. S. Watson, “Spectral-temporal factors in the identification of environmental sounds,” *The Journal of the Acoustical Society of America*, vol. 115, no. 3, pp. 1252–1265, 2004.

[35] J. J. Ohala, “Cross-language use of pitch: an ethological view,” *Phonetica*, vol. 40, no. 1, pp. 1–18, 1983.

[36] C. Gussenhoven, “Intonation and interpretation: phonetics and phonology,” in *Proceedings of Speech Prosody 2002, International Conference*, 2002, pp. 47–57.

[37] P. N. Juslin, “Cue utilization in communication of emotion in music performance: relating performance to perception.” *Journal of Experimental Psychology: Human perception and performance*, vol. 26, no. 6, p. 1797, 2000.

[38] A. Amador and D. Margoliash, “A mechanism for frequency modulation in songbirds shared with humans,” *The Journal of Neuroscience*, vol. 33, no. 27, pp. 11 136–11 144, 2013.

The Effect of DJs' Social Network on Music Popularity

Hyeongseok Wi
Korea Advanced Institute of Science and Technology
trilldogg@kaist.ac.kr

Kyung Hoon Hyun
Korea Advanced Institute of Science and Technology
hellohoon@kaist.ac.kr

Jongpil Lee
Korea Advanced Institute of Science and Technology
richter@kaist.ac.kr

Wonjae Lee
Korea Advanced Institute of Science and Technology
wnjlee@kaist.ac.kr

ABSTRACT

This research focuses on two distinctive determinants of DJ popularity in Electronic Dance Music (EDM) culture. While one's individual artistic tastes influence the construction of playlists for festivals, social relationships with other DJs also have an effect on the promotion of a DJ's works. To test this idea, an analysis of the effect of DJs' social networks and the audio features of popular songs was conducted. We collected and analyzed 713 DJs' playlist data from 2013 to 2015, consisting of audio clips of 3172 songs. The number of cases where a DJ played another DJ's song was 15759. Our results indicate that DJs tend to play songs composed by DJs within their exclusive groups. This network effect was confirmed while controlling for the audio features of the songs. This research contributes to a better understand of this interesting but unique creative culture by implementing both the social networks of the artists' communities and their artistic representations.

1. INTRODUCTION

Network science can enhance the understanding of the complex relationships of human activities. Thus, we are now able to analyze the complicated dynamics of sociological influences on creative culture. This research focuses on understanding the hidden dynamics of Electronic Dance Music (EDM) culture through both network analysis and audio analysis.

Disc Jockeys (DJs) are one of the most important elements of EDM culture. The role of DJs is to manipulate musical elements such as BPM and timbre [1] and to create unique sets of songs, also known as playlists [2]. DJs are often criticized on their ability to "combine" sets of songs, since the consistency of atmosphere or mood is influenced by the sequence of the songs [3]. Therefore, it is common for DJs to compose their playlists with songs from other DJs who share similar artistic tastes. However, there are other reasons aside from artistic tastes that contribute to a DJ's song selection. DJs sometimes strategically play songs from other DJs because they are on the same record labels; thus, playlist generation is influenced by a complex mixture of artistic and social reasons. This interesting dynamic of EDM culture has led us to ask two specific questions: What reasons are most important for DJs when selecting songs to play at a festival? How do social relationships or audio features influence the popularity of songs? By answering these two questions, we can better understand the mechanisms of how DJs gain popularity and how their artistic tastes influence the construction of playlists for festivals.

To answer the above, we conducted the following tasks: 1) DJ networks based on shared songs were collected; 2) Audio data of the songs played by the DJs were collected; 3) Network analysis was conducted on DJ networks; 4) Audio features were extracted from the collected audio data; 5) The relationships between DJ networks and audio features were identified through three longitudinal Fixed Effect Models.

2. RELATED WORKS

2.1 Social Networks of Musicians

Network analysis has been widely applied to the field of sociology and physics. Recently, researchers have started adopting network analysis to better understand the underling mechanisms of art, humanities and artists' behavior. Among the few attempts to implement network analysis in the field of music, researchers have tried to investigate how musicians are connected to other musicians in terms of artistic creativity.

The effects of collective creation and social networks on classical music has been previously studied. McAndrew et al. [4] analyzed the networks of British classical music composers and argued that it is conceptually difficult to separate music from its social contexts. This is because it is possible for creative artworks to be influenced by musicians' social interactions and collaborations, and, moreover, an artist's intimate friendships can even create his or her own styles and artistic innovations.

Gleiser and Danon [5] conducted research on racial segregation within the community of jazz musicians of the 1920's through social interaction network analysis. Park et al. [6] analyzed the properties of the networks of western classical music composers with centrality features. The results of this analysis showed small world network characteristics within the composers' networks. In addition, composers were clustered based on time, instrumental positions, and nationalities. Weren [7] researched collegiate marching bands and found that musical performance and motivation were higher when musicians were more integrated into a band's friendship and advice networks.

It is widely known that the most important elements of artistic communities are individuals' creativity and novelty. However, the literature on the social networks of musicians argues that the social relationships of artists are important elements within creative communities as well.

2.2 Audio Computing

There are various feature representations in the field of Music Information Retrieval (MIR) [8]. Since the goal of the research is to find the influence of DJs' social relationships and their artistic tastes on music popularity, it is important to extract audio features that consist of rich information. Timbre is one of the most important audio features when DJs create playlists [1]. Additionally, tonal patterns are equally important in EDM songs [9]. Therefore, we extracted Mel-frequency cepstral coefficients (MFCC), Chroma, tempo and Root-Mean-Square Energy (RMSE) to cover most musical characteristics such as musical texture, pitched content and rhythmic content [10]. Beat synchronous aggregation for MFCC, Chroma and RMSE was applied to make features more distinctive [11]. The harmonic part of the spectrograms were used for Chroma, and the percussive part of the spectrograms were used for beat tracking by using harmonic percussive separation [12]. After the features were extracted, the mean and standard deviations of MFCC, Chroma and RMSE were taken to supply a single vector for each song [1]. All audio feature extraction was conducted with librosa [12].

3. HYPOTHESIS

DJs not only creatively construct their own playlists to express their unique styles, but also manipulate existing songs to their artistic tastes. This process is called remixing. DJs remix to differentiate or familiarize existing songs for strategic reasons. Therefore, the songs are the fundamental and salient elements of EDM culture. For this reasons, DJs delicately select songs when constructing playlists to ultimately satisfy universal audiences' preferences. Thus, the frequency of songs selected by DJs represents the popularity of the songs. Thus, the logical question to ask is, "What are the most important factors when DJs select songs?" Our hypotheses based on this question are as follows:

H1. Song popularity would correlate with DJs' artistic tastes, controlling for the social relationships of DJs.

H2. The social relationships of DJs would influence song popularity, controlling for DJs' artistic tastes.

4. METHODOLOGY

Songs' popularity were calculated based on DJ network, while audio features were extracted from audio clips of the songs. As a result, we collected and extracted DJ network data and audio clips. Ultimately, the dynamics of DJ networks and audio features were analyzed through the Fixed Effect Model.

4.1 Data Set

We collected 713 DJs' playlist data (from 2013 to 2015) through Tracklist.com (from a total of 9 notable festivals: Amsterdam Dance Event (Amsterdam); Electric Daisy Carnival (global); Electric Zoo (US); Mysteryland (global); Nature One (Germany); Sensation (global); Tomorrow-Land (Belgium); Tomorrowworld (US); Ultra Music Festival (global)); and audio clips from Soundcloud.com (within license policies)).

Three types of data were constructed based on the collected data: 1) networks of DJs playing other DJs' songs; 2) popularity of the songs by calculating the frequencies of songs played at each festival; and 3) audio features from audio clips, filtering out audio clips that were shorter than 2 minutes long. To summarize, playlist networks and audio clips of 3172 songs with 15759 edges were collected and analyzed.

4.2 DJ Network Analysis

As shown in Figure 1, DJ networks were constructed based on directed edges. When DJ_1 plays a song composed by DJ_2 and DJ_3, we consider DJ_1 as having interacted with DJ_2 and DJ_3.

The DJ networks consisted of 82 festivals that were merged down to 77 events due to simultaneous dates. Therefore, we constructed 77 time windows of DJ interaction (play) networks based on festival event occurrence. A song's popularity was calculated based on the number of songs played in each time window. We also calculated the betweenness centrality, closeness centrality, in-degree and out-degree of DJs.

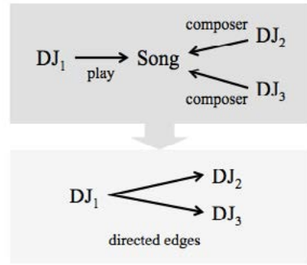


Figure 1. Construction of DJ Networks

The betweenness centrality of a node reflects the brokerage of the node interacting with other nodes in the network. For instance, a higher betweenness centrality signifies that the nodes connect different communities. A lower betweenness centrality indicates that the nodes are constrained within a community. Closeness centrality represents the total geodesic distance from a given node to all other nodes. In other words, both higher betweenness and closeness centralities indicate that the DJs tend to select songs of various DJs. Lower betweenness and closeness

centralities signify that the DJs tend to select songs within the same clusters. In-degree is the number of a DJ's songs played by other DJs. Out-degree is the number of a DJ's play count of other DJs' songs.

4.3 Audio Analysis

We extracted audio features related to tempo, volume, key and timbre from 3172 songs. The sequential features are collapsed into mean and standard deviation values to maintain song-level value and dynamics [1]. A total of 52 dimensions are used, including tempo (1), mean of RMSE (1), mean of Chroma (12), mean of MFCC13 (13), standard deviation of Chroma (12) and standard deviation of MFCC13 (13).

5. IMPLEMENTATIONS & RESULTS

We fit a longitudinal fixed effects model:

$$Y_{k,t+1} = Y_{k,t} + \mathbf{S}_k\phi + \mathbf{W}_{ij,t}\beta + \mu_k + \tau_t + e_{k,t+1} \qquad (1)$$

, where the dependent variable, $Y_{k,t+1}$ is the frequency of a song k that was played in the event t+1. $Y_{k,t}$ is the lagged dependent variable (t). By including the lagged dependent variable, we expect to control for "mean reversion" and self-promotion effect.

μ_k is a vector of the fixed effects for every song k. By including this, the time-invariant and song-specific factors are all controlled. For example, the effects of the composer, the label, and the performing artists are all controlled for with μ_k.

τ_t is a vector of time fixed effects. Each song is assumed to be played at a particular time whose characteristics such as weather and social events would have an exogenous effect on $Y_{k,t+1}$. τ_t controls for the unobserved heterogeneity specific to the temporal points.

$\mathbf{S}_k$ is the vector of the song k's audio features which include the average and standard deviation of Chroma, MFCC, RMSE, and tempo. The value of audio features is time-invariant and, therefore, perfectly correlated with the fixed effects (μ_k). To avoid perfect collinearity with the fixed effects, we quantize the values into five levels, and make a five-point variable for each characteristic.

$\mathbf{W}_{ij,t}$ is the vector of the network covariates. Network centralities of DJ i who composed k are calculated using a network at time t. In the network matrix, the element w_{ij} is the frequency i played j's song at time t.

This research focuses on two distinctive determinants of DJ popularity in Electronic Dance Music (EDM) culture. While a DJ's individual artistic tastes influence the construction of playlists for festivals, social relationships with other DJs also have an effect on the promotion of a DJ's works. To test this idea, an analysis of the effect on song popularity by DJ social networks and song audio features was conducted. Song popularity among DJs was used as a dependent variable. We conducted three different Longitudinal Fixed Effect Models. Model 1 finds the influence

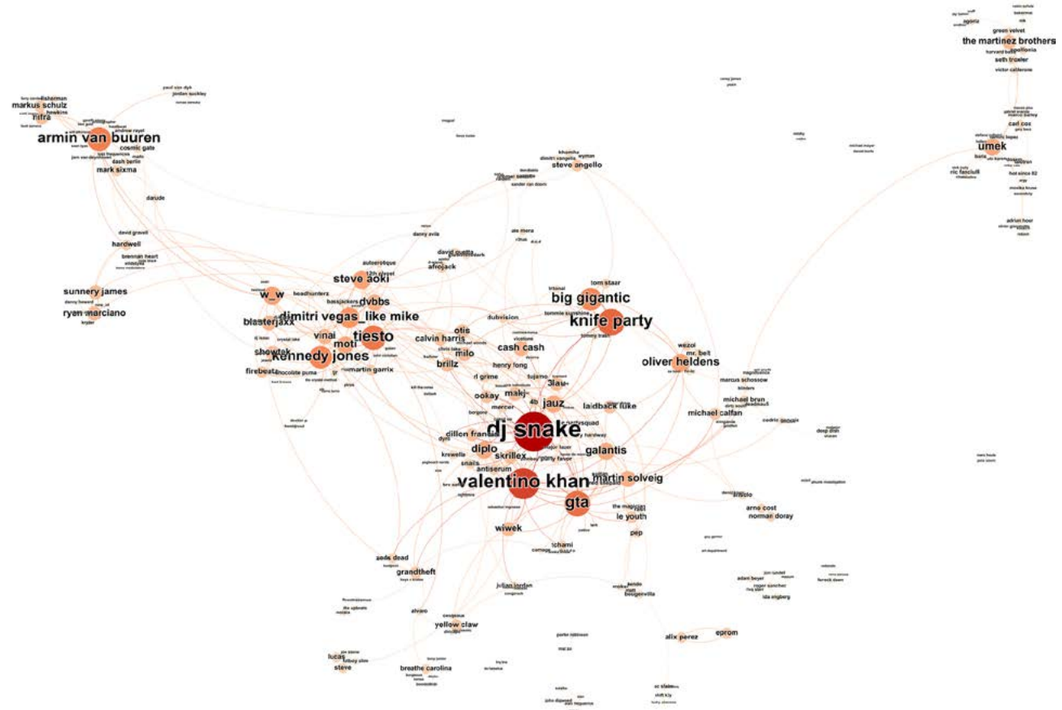


Figure 2. Backbone Network Graph of Ultra Music Festival 2015 Miami.

of audio features on song popularity, and Model 2 determines the effect of social relationships on song popularity. In this case, social relationship information such as betweenness, closeness, in-degree and out-degree were used as independent variables when audio features such as RMSE, tempo, Chroma and MFCC were used as control variables. This analysis was based on 77 different time windows. For Model 3, we combine Model 1 and Model 2, controlling the audio features and social relationships on song popularity.

Model 1 shows stable results indicating the presence of shared audio features within DJ networks (Appendix 1). In particular, the mean of Chroma 10 negatively correlated with song popularity ($p < 0.001$). Chroma 10 represents "A" pitch, which can be expressed as "A" key. Considering that song popularity is calculated based on DJs playing other DJs' songs, this result suggests that DJs tend to avoid using "A" key when composing songs. Therefore, we can argue that commonly shared artistic tastes exist. However, artistic tastes will continue to change depending on trends. Further study is needed to better interpret the relationships between audio features and song popularity (Table 1).

Popular songs	Popularity	Chroma10
W&W – The Code	92	0.3003
Hardwell - Jumper	104	0.3012
Blasterjaxx – Rocket	84	0.2890
Martin Garrix – Turn Up The Speaker	88	0.2349
Markus Schulz	52	0.2201

Table 1. Example of songs' popularity and Chroma 10, (mean of entire song's Chroma 10 = 0.3770; mean of entire songs' popularity = 7.4943)

On the other hand, social networks of DJs are expected to be more consistent than artistic tastes. Based on Model 2, the effect of DJ social relationships on song popularity showed firm stability (Appendix 1). Based on Model 3, audio features and DJ social networks independently influence song popularity. Despite socially biased networks of DJs, DJs appeared to have shared preferences on audio features within their clusters. Table 2 shows negative correlations of song popularity on both betweenness ($p < 0.05$) and closeness ($p < 0.001$) of DJ networks. In other words, the more popular a song is, the more often the song is played within the cluster (Figure 4).

Variables	Coefficients
Song Popularity	0.112*** (0.011)
In-Degree	-0.001 (0.001)
Out-Degree	-0.000 (0.001)
Closeness	-0.382*** (0.079)
Betweenness	-0.000* (0.000)
Constant	-3.274 (2.296)

Table 2. The Result of the Fixed Effect Model (Standard Errors in Parentheses; *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$)

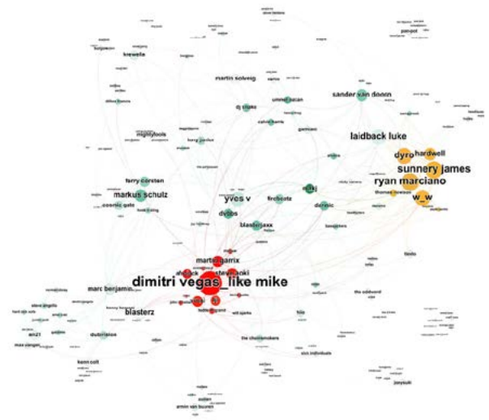


Figure 4. Composers of popular songs colored within the DJs clusters. (Tomorrowland 2014, Belgium)

Based on this result we can conclude that DJs tend to play songs composed by DJs from their exclusive groups independently from audio features. To conclude, H1 is supported by Models 1 and 3. H2 is supported by Models 2 and 3.

6. CONCLUSION

This research focuses on understanding the mechanism of artistic preferences among DJs. The artistic preferences of universal audiences are not considered in this research. Thus, the network cluster effect shown in this research needs to be considered as a social bias effect among DJs' artistic collaboration networks rather than the popularity of universal audiences. However, the result of the research shows that DJs tend to prefer DJs who are centered within their clusters. Therefore, the social networks of DJs influence on their song selection process.

The contributions of this research are as follows. Firstly, creative culture consists of complex dynamics of artistic and sociological elements. Therefore, it is important to consider both the social networks of artist communities and their artistic representations to analyze creative culture. Secondly, the proposed research methodology can help to unveil hidden insights on DJs' creative culture. For instance, DJs have unique nature of composing new songs by manipulating and remixing existing songs created by themselves or other DJs. Burnard [14] stated that the artistic creativity is often nurtured by artists who build on each other's ideas by manipulating the existing artworks. The understanding of this interesting collaborative culture can unveil novel insights on creative collaboration.

For future works, we will research the mechanism of artistic preferences of universal audiences along with DJs' collaboration networks. In addition, more detailed research on the effects of audio features on each cluster can provide deeper insights on understanding EDM culture. By analyzing the networks of DJs' remixing behavior and state of the art audio analysis, we can further investigate the clusters of DJs' artistic tastes and their collaboration patterns.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government(MSIP) (R0184-15-1037) and National Research Foundation (NRF-2013S1A3A2055285).

7. REFERENCES

[1] T. Kell and G. Tzanetakis, "Empirical Analysis of Track Selection and Ordering in Electronic Dance Music using Audio Feature Extraction," *ISMIR*, 2013.

[2] T. Scarfe, M.Koolen and Y. Kalnishkan, "A long-range self-similarity approach to segmenting DJ mixed music streams," *Artificial Intelligence Applications and Innovations*, Springer Berlin Heidelberg, p. 235-244, 2013.

[3] B. Attias, A. Gavanas and H. Rietveld, *"DJ culture in the mix: power, technology, and social change in electronic dance music"*, Bloomsbury Publishing USA, 2013.

[4] S. McAndrew and M. Everett, "Music as Collective Invention: A Social Network Analysis of Composers," *Cultural Sociology,* vol.9, no.1, pp. 56-80, 2015.

[5] P.M. Gleiser and L. Danon, "Community structure in jazz," *Advances in complex systems,* vol. 6, no.04, pp. 565-573, 2005.

[6] D. Park, A. Bae and J. Park, "The Network of Western Classical Music Composers," *Complex Networks V, Springer International Publishing*, p. 1-12, 2014.

[7] S. Weren, "Motivational and Social Network Dynamics of Ensemble Music Making: A Longitudinal Investigation of a Collegiate Marching Band", *Diss*, Arizona State University, 2015.

[8] M. Casey, A. Michael, R. Veltkamp, R., M. Goto, R.C. Leman, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE,* vol. 96, no. 4 pp. 668-696, 2008.

[9] R.W. Wooller and R.B. Andrew. "A framework for discussing tonality in electronic dance music," 2008.

[10] J. Paulus, M. Müller, and A. Klapuri, "State of the Art Report: Audio-Based Music Structure Analysis," *ISMIR*, 2010.

[11] D. PW. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research,* vol. 36, no.1, pp. 51-60, 2006.

[12] D. Fitzgerald, "Harmonic/percussive separation using median filtering," 2010.

[13] B. McFee, "librosa: Audio and music signal analysis in python, " *Proceedings of the 14th Python in Science Conference*, 2015.

[14] P. Burnard and M. Fautley, "Assessing diverse creativities in music," *The Routledge International Handbook of the Arts and Education*, Routledge, p.254-267, 2015.

APPENDIX

VARIABLES	Model (1)	(2)	(3)
Chroma_mean1_quint	0.279 (0.208)		0.277 (0.208)
Chroma_mean2_quint	0.081 (0.077)		0.081 (0.077)
Chroma_mean3_quint	-0.003 (0.016)		-0.004 (0.016)
Chroma_mean4_quint	-0.306* (0.125)		-0.305* (0.124)
Chroma_mean5_quint	-0.043*** (0.012)		-0.044*** (0.013)
Chroma_mean6_quint	0.525* (0.242)		0.528* (0.242)
Chroma_mean7_quint	0.008 (0.009)		0.007 (0.009)
Chroma_mean8_quint	-0.008 (0.019)		-0.009 (0.019)
Chroma_mean9_quint	-0.105 (0.063)		-0.106 (0.063)
Chroma_mean10_quint	-0.089*** (0.011)		-0.090*** (0.011)
Chroma_mean11_quint	-0.052* (0.021)		-0.053* (0.021)
Chroma_mean12_quint	0.096 (0.111)		0.097 (0.112)
Chroma_std1_quint	-0.019 (0.033)		-0.019 (0.032)
Chroma_std2_quint	-0.252 (0.177)		-0.253 (0.177)
Chroma_std3_quint	-0.016 (0.140)		-0.015 (0.139)
Chroma_std5_quint	-0.042*** (0.009)		-0.042*** (0.008)
Chroma_std6_quint	0.729* (0.330)		0.725* (0.328)
Chroma_std7_quint	-0.004 (0.030)		-0.003 (0.030)
Chroma_std8_quint	0.653* (0.306)		0.650* (0.306)
Chroma_std9_quint	-0.043* (0.022)		-0.044* (0.023)
Chroma_std10_quint	-0.185*** (0.035)		-0.185*** (0.034)
Chroma_std11_quint	-0.036 (0.075)		-0.035 (0.075)
Chroma_std12_quint	0.340 (0.205)		0.338 (0.204)
MFCC_mean1_quint	-0.059*** (0.016)		-0.058*** (0.015)
MFCC_mean2_quint	0.078 (0.194)		0.079 (0.193)
MFCC_mean3_quint	-0.169*** (0.048)		-0.167*** (0.047)
MFCC_mean4_quint	-0.032 (0.020)		-0.032 (0.021)
MFCC_mean5_quint	0.108 (0.055)		0.108 (0.056)
MFCC_mean6_quint	0.020 (0.044)		0.019 (0.042)
MFCC_mean7_quint	0.046 (0.116)		0.046 (0.115)
Continued (…)	Continued (…)	Continued (…)	Continued (…)

Continued (…)	Continued (…)	Continued (…)	Continued (…)
MFCC_mean8_quint	0.050 (0.049)		0.051 (0.048)
MFCC_mean9_quint	0.148* (0.076)		0.147* (0.075)
MFCC_mean10_quint	-0.097 (0.069)		-0.097 (0.069)
MFCC_mean11_quint	0.016 (0.014)		0.015 (0.015)
MFCC_mean12_quint	-0.087* (0.036)		-0.089* (0.035)
MFCC_mean13_quint	0.005 (0.011)		0.005 (0.011)
MFCC_std1_quint	-0.042 (0.035)		-0.043 (0.034)
MFCC_std2_quint	0.034 (0.086)		0.035 (0.086)
MFCC_std3_quint	-0.053 (0.047)		-0.052 (0.046)
MFCC_std4_quint	0.038 (0.037)		0.038 (0.037)
MFCC_std5_quint	-0.100** (0.031)		-0.100** (0.032)
MFCC_std6_quint	-0.318** (0.097)		-0.318** (0.096)
MFCC_std7_quint	-0.103 (0.071)		-0.104 (0.070)
MFCC_std8_quint	0.141** (0.053)		0.142** (0.054)
MFCC_std9_quint	-0.053* (0.022)		-0.052* (0.024)
MFCC_std10_quint	0.175 (0.106)		0.177 (0.107)
MFCC_std11_quint	-0.017 (0.014)		-0.018 (0.014)
MFCC_std12_quint	0.017 (0.011)		0.019 (0.012)
MFCC_std13_quint	0.020 (0.025)		0.020 (0.025)
RMSE_mean_quint	0.005 (0.019)		0.005 (0.019)
Tempo_quint	0.010 (0.006)		0.010 (0.006)
Song popularity	0.109*** (0.011)	0.114*** (0.011)	0.112*** (0.011)
In_degree		-0.001 (0.001)	-0.001 (0.001)
Out_degree		-0.000 (0.001)	-0.000 (0.001)
Closeness Centrality		-0.383*** (0.079)	-0.382*** (0.079)
Betweenness Centrality		-0.000* (0.000)	-0.000* (0.000)
Constant	-3.853 (2.298)	0.567*** (0.115)	-3.274 (2.296)
Song fixed effects	Yes	Yes	Yes
Time fixed effects	Yes	Yes	Yes
Observations	241,072	241,072	241,072
R-squared	0.1335	0.1323	0.1338
Adjusted R-squared	0.1214	0.1205	0.1218
Number of id	3,172	3,172	3,172

Robust standard errors in parentheses
*** p<0.001, ** p<0.01, * p<0.05

Appendix 1. Fixed Effect Model for Model (1), (2) and (3).

Kronos Meta-Sequencer – From Ugens to Orchestra, Score and Beyond

Vesa Norilo
Centre for Music & Technology
University of Arts Helsinki
vno11100@uniarts.fi

ABSTRACT

This article discusses the Meta-Sequencer, a circular combination of an interpreter, scheduler and a compiler for musical programming. Kronos is a signal processing language focused on high computational performance, and the addition of the Meta-Sequencer extends its reach upwards from unit generators to orchestras and score-level programming. This enables novel aspects of temporal recursion – a tight coupling of high level score abstractions with the signal processors that constitute the fundamental building blocks of musical programs.

1. INTRODUCTION

Programming computer systems for music is a diverse practice; it encompasses everything from fundamental synthesis and signal processing algorithms to representing scores and generative music; from carefully premeditated programs for tape music to performative live coding.

One estabilished classification of musical programming tasks, arising from the MUSIC-N tradition [1, pp. 787-796] [2], identifies three levels of abstraction:

1. Unit Generator
2. Orchestra
3. Score

Unit Generators are the fundamental building blocks of musical programs, including oscillators, filters and signal generators. Orchestras are ensembles of Unit Generators, coordinated to behave as musical instruments. Finally, scores encode control information – a high level representation of a piece to be performed by the Unit Generator Orchestra. Most MUSIC-N family languages are based on distinct domain languages for orchestras and scores; programming unit generators from scratch is rarely addressed.

This paper addresses the problem of tackling all three levels of hierarchy in a single programming language. It is based on extending Kronos [3], a functional reactive signal processing language, with the notion of a task scheduler and a script interpreter capable of driving each other. This notion enables a powerful expression of score metaphors, including temporal recursion [4]. This is the concept of the Meta-Sequencer – a programmable sequencer capable of reprogramming itself.

2. BACKGROUND

Contemporary musical programming languages often blur the lines between ugen, orchestra and score. Semantically, languages like Max [5] and Pure Data [6] would seem to provide just the Orchestra layer; however, they typically come with specialized unit generators that enable score-like functionality. Recently, Max has added a sublanguage called Gen to address ugen programming.

SuperCollider [7] employs an object-oriented approach of the SmallTalk tradition to musical programming. It provides a unified idiom for describing orchestras and scores via explicit imperative programs.

ChucK [8] introduces timing as a first class language construct. ChucK programs consist of Unit Generator graphs with an imperative control script that can perform interventions at precisely determined moments in time. To draw an analogy to MUSIC-N, the control script is like a score – although more expressive – while the unit generator graph resembles an orchestra. The ChucK model can also extend to "natively constructed" ugens [8, pp. 25-26].

Languages specifically focused on ugens are both fewer and more recent. Faust [9] is a prominent example, utilizing functional programming and block diagram algebra to enable compact descriptions of unit generators while maintaining high computational efficiency. The functional model is a good fit for computationally efficient signal processing: its traits, such as immutable values, referential transparency and suitability for equational reasoning [10] enable a high degree of compiler optimization. My own prior work with Kronos [3] is inspired by the Faust model, seeking to contribute mixed-rate and event-driven systems as well as type-based polymorphism and metaprogramming.

The problem of combining all three levels in a single language is challenging yet intriguing. Successful orchestra/score languages like Max and ChucK have some facilities for ugen programming [8, pp. 25-26]. The respective tradeoffs include semantics that differ from the rest of the environment, and computational efficiency far below machine limits. Brandt has studied ugen-type programming with temporal type constructors [11] and related tradeoffs, such as limitations in program semantics and a lack of real-time capabilities.

This study approaches the problem from the opposite direction: extending Kronos [3], a signal-processing, ugen-orchestra-focused language upward to provide score capability. This is achieved by a novel, embedded domain language inspired by the I/O Monad in Haskell [12] and the concept of temporal recursion [4].

3. META-SEQUENCER

The project that became the Kronos Metasequencer originated in an effort to improve the expressibility of outputs in Kronos programs. While functional programming [13] excels in expressing data flows and signal topologies, it is less suitable for modeling any *effects* the program should have on its surroundings. Functional programs do not encode state, but state is present in all the relevant input/output devices attached to computers. Programs must mutate that state in order to be observable (and potentially useful) to their users.

One approach to the problem is to externalize the I/O concerns. Faust [9] programs are assumed to output an audio stream. As Kronos [3] extends signal processing to event models such as MIDI and OSC [14], a more complicated solution is required. The specification of the signal destination can still be externalized: whether a program should output audio, OSC events, MIDI messages or text onto a console could be supplied to the compiler as additional parameters, not indicated in the source code in any way.

The second, more refined approach could involve type polymorphism: an appropriate output destination would be determined based on the data type of the output. Programs could specify the desired output behavior simply by returning a type such as a MIDI event. While this approach has benefits, the set of types becomes complicated as the variety of output methods grows. In addition, the compiler driver must interpret all the types: the output specification quickly starts to resemble a mini-language of its own.

3.1 Haskell and I/O

It is instructive to look at how a pure, general purpose functional language like Haskell [12] encodes I/O operations. At first glance, I/O code in Haskell appears imperative: the syntax evokes assignment and side-effectful read and write operations. A simple example from the Haskell wiki is shown in Listing 1. Despite the code appearing imperative, functional purity is not compromised: referential transparency [10] and equational reasoning remain in force.

Listing 1. Simple Haskell program with I/O

```
main = do a <- ask "What is your name?"
          b <- ask "How old are you?"
          return ()

ask s = do putStr s
           readLn
```

This implementation is powered by the I/O monad, which provides a functional representation of an imperative program, modeling the stateful effects caused by I/O actions as an implicit data flow. Monadic I/O code is a domain language within Haskell.

3.2 The I/O Domain Language

Various domain languages built in Kronos already exist; these range for small-scale experiments such as document generation to published work on graphics and animation [15]. This section describes a domain language for I/O, capable of enabling semantics such as evoked by Listing 2.

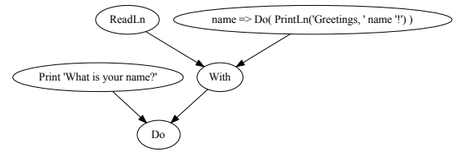


Figure 1. Abstract Syntax Tree of an Imperative Kronos Program

Listing 2. Kronos program with I/O

```
Greetings() {
  Use Actions
  Greetings = Do(
     Print("What is your name? ")
     name <- ReadLine()
     PrintLn("Greetings, " name "!") )
}
```

3.2.1 The Imperative Interpreter

As metaprogramming is one of Kronos' fundamental principles [3], the I/O domain language is based on the concept of second-order code generation: a functional dataflow program constructing the syntax tree of an imperative program. Data types are used to encode the *abstract syntax tree* (AST) of the imperative program. For example, the AST generated by running Listing 2 is shown in Figure 1.

Similar to how I/O Actions in Haskell are effective only at the root of the program entry point, the I/O language is designed around an interpreter hook placed at the very end of the program data flow. It is implemented as a foreign-function call to the compiler driver written in C++. The interpreter will then traverse the AST, executing the effectful Print and ReadLn nodes.

3.2.2 Enabling Assignment Semantics

There is a non-obvious detail in the AST shown in Figure 1: the left-arrow syntax, simulating assignment, has been translated to a node called 'With'. This node receives the I/O action whose result value is bound to 'name', and a closure encompassing the remaining I/O actions that were sequenced after it. The AST interpreter will invoke the I/O action passed as the value, and invoke the closure with the result of the I/O action as a parameter. If the result of the closure is another I/O action, the interpreter will then process that action.

To illustrate the assignment transformation, Listing 3 shows how Listing 2 is lowered.

Listing 3. Example of Assignment Transformation

```
; after left-arrow transformation
Do(
  Print("What is your name? ")
  Invoke-With(
     ReadLine()
     name => PrintLn("Greetings, " name "!")))
```

3.2.3 The Interpreter as Compiler Driver

It is noteworthy that the closures shown in Listing 3 are constructs of the core Kronos language. The interpreter

does not know how to compute: all numeric work is delegated to the dataflow compiler. This includes generation of the interpreter ASTs: the closure shown in the transformed version actually returns an imperative program to print 'name'.

Execution of programs with actions is essentially a cycle of interpretation of an imperative AST, alternating with compilation and execution of pure functional code, which may produce a new imperative AST.

For performance reasons, Kronos programs are statically typed. However, the Kronos methodology is based on type-generic programs: there are very rarely any type annotations in the source code, as the compiler will estabilish the types through whole-program type derivation.

As the interpreter drives compilation, also providing the root types for the compiler, it effectively appears to the user as a dynamically typed language, where type-specific routines are compiled on demand.

3.2.4 Control Flow

An important aspect of any imperative scripting is control flow – decision points where the program flow could diverge based on run-time conditions. While such control flow is highly toxic to high performance signal processing, it is essential for many score-level tasks.

For these purposes, the imperative language contains an If-node, structured in the well-known format of truth value – then-branch – else-branch. The AST interpreter will retrieve the truth value and based on it, proceed on to either the then-branch or the else-branch.

Please recall that 'If' is a normal function. That means, on one hand, that it can be used in the variety of ways functions can: composed, applied partially, passed as a parameter value, and so on. On the other hand, data flow demands that all of its upstream children, truth-value, then-branch and else-branch must be evaluated prior to it.

With this in mind, please consider a looping structure, such as shown in Listing 4. Because one side of the conditional branch refers recursively back to itself, a straightforward implementation of 'If' would result in an infinitely deep AST and nontermination. A common strategy to address this problem in functional programming is lazy evaluation [13, p. 384], while imperative languages favor short-circuiting or minimal evaluation. Both require specific support from the compiler.

Listing 4. Recursion and Control Flow

```
Countdown(count) {
  Use Actions
  Countdown =
    If(count > 0
       { Do(
          PrintLn(count)
          Countdown(count - 1)) }
       { PrintLn("Done") })
}
```

Kronos can support a rudimentary form of explicit lazy evaluation by specifying that the then-branch and the else-branch are in fact closures, and should return the AST to be taken by the interpreter. As anonymous functions can be written simply by enclosing statements in curly braces, the resulting syntax should be quite familiar to programmers. An example of a recursive looping program is shown in Listing 4. Without the intermediate closures, this program would fail by getting stuck in an infinite loop in the AST generation stage.

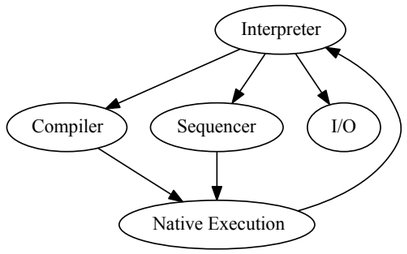


Figure 2. Meta-Sequencer Program Flow

3.3 Temporal Recursion: Meta-Sequences

So far, the imperative language features discussed in this paper have little relevance to computer music, as they are little more than the staples of imperative programming defined in a functional dataflow language designed for DSP.

However, a simple addition to the interpreter-compiler-execution cycle will bring about a significant expansion to musical possibilities. Kronos already features a sequencer for timed reactive events [16]. Extending that sequencer to schedule and fire imperative programs is a logical evolution. If, in addition, the imperative programs gain facility to program the sequencer, the expressive power of the system grows significantly.

This is the concept of the Meta-Sequencer: a fusion of an interpreter, a sequencer and a compiler . The program flow is shown in Figure 2. The interpreter traverses an AST, directing it to fire I/O events or to compile a Kronos function for execution either directly or as scheduled by a sequencer. The compiled function may contain an interpreter hook, cycling back to the interpreter for further actions.

In fact, this closely follows the concept of temporal recursion as presented by Sorensen [4], as well as other declarative methods in the literature [17, 11, 18].

3.4 I/O Actions in Detail

This section summarizes the imperative I/O Action language and the primitives of its AST, which are displayed in Table 1.

3.4.1 After

After is the scheduling command. It can be used to schedule an arbitrary AST for execution after a specified period in seconds. Scheduling is sample-accurate and synchronous with the audio stream.

3.4.2 Send

Send represents a discrete output event. The arguments to this command are an address pattern and value. The address pattern determines the output method. An URI-type scheme is used here: for example, OSC [14] outputs can be specified by "osc://ip:port/osc/address/pattern". The *Print* command utilizes *Send*, specifying an address pattern reserved for console output.

While arbitrary values can be passed to *Send*, the output method may not be able to handle all data types. The OSC encoder can handle primitive numbers, strings, truth values as scalars and nested arrays, but more complicated types such as closures are not supported.

Table 1. I/O Actions in Kronos

Action	Arguments	Description
After	*time fn*	*run 'fn' 'time' secs later*
Do	*actions*	*run 'actions' sequentially*
For	*values fn*	*apply 'fn' to each element in 'values'*
Invoke-With	*action fn*	*Pass result of 'action' to 'fn'*
If	*p t e*	*If 'p' is true, invoke 't'hen to obtain a new AST; else invoke 'e' for it.*
Send	*address value*	*Output 'value' to 'address'*
Send-To	*id addr val*	*Send 'val' to method 'addr in instance 'id'*
Print	*value*	*Send("#pr" value)*
PrintLn	*value*	*Do(Print(value) Print("\n"))*
ReadLine		*Read line from console and return as a string*
Start	*fn*	*Start 'fn' as a reactive instance return an instance id.*
Stop	*id*	*Stops the instance 'id'*

3.4.3 Start, Stop, Send-To

Start instantiates a Kronos closure as a *reactive object*, responding to reactive inputs and producing a stream of outputs. Each instance is a *discrete reactive system* according to the classification presented by Van Roy [19].

The return value of the *Start* command is an instance handle. The referred instance can be stopped by passing the handle to *Stop*. This can be done by the top-level REPL or any script that fires within the sequencer. The handle is also passed to the closure itself, enabling it to stop itself.

Send-To is a convenience function that works like *Send*, but addresses an input within a specific instance identified by a handle.

4. APPLICATIONS AND IMPLICATIONS

4.1 Reactive Event Processors

The Kronos signal model is based on reactive update propagation [16]. The imperative ASTs participate in this signal model – if a reactive signal feeds a leaf of the AST, it effectively becomes an event handler for that signal. This results in a very simple definition of an OSC [14] monitor, shown in Listing 5.

Listing 5. Reactive OSC Monitor

```
; listen to float values at OSC address pattern '/a'
Start( { PrintLn(Control:Param("/a" 0)) } )
```

This instance will print a line of text representing each OSC method call that supplies a floating point value to address "/a". Signal-flow-wise, *Control:Param* returns a float scalar, which *PrintLn* translates into an imperative program to print said scalar. This is in turn sent to the interpreter via the interpreter hook implicitly placed at the root of the closure. Reactivity flows downstream from *Control:Param*, so the interpreter hook fires whenever there is OSC input.

4.2 Generative Sounds

The next example, Generative Sound, utilizes temporal recursion to construct a sonic fractal. The code is shown in Listing 6. The fractal plays a sinusoid for a specified duration, spawning delayed, recursive copies of itself to generate increasingly dense partials.

Listing 6. Sonic Fractal

```
Import Gen

Fractal(f dur g) {
  Use Actions

  ; time offset to next cluster
  time-offset = Math:Sqrt(dur)

  ; its duration is the remaining time
  next-dur = dur - time-offset

  Fractal = Do(
     ; start sinusoid at frequency 'f'
     id <- Start( { Wave:Sin(f) * g } )

     ; stop it after 'dur' seconds
     After( dur Stop( id ) )

     ; spawn two more fractals at musical intervals
     ; of 2/3 and 6/5, after time offset
     If( dur > 0.5
       { After( time-offset
         Fractal(f * 2 / 3 next-dur g / 2)
         Fractal(f * 6 / 5 next-dur g / 2) ) } )
  )
}
```

The fractal could be made more musically interesting with features such as randomized offsets or additional timbre parameters. Even the simple form demonstrates the generative power of temporal recursion.

An additional benefit of the fractal is benchmarking: recall that the control script is both sample accurate and audio-synchronous; in real-time playback, this has a significant computational impact when a high number of closures are scheduled to be compiled and played back at once.

In informal benchmarking, constructing and connecting an instance (after initial warm-up) in Listing 6 happens in $30\mu s$ on a laptop with Intel Core i7-4500U processor. Sinusoid synthesis is computationally cheap, so instantiation is the main constraint on real-time playback. As the fractal features 2^N sinusoids at step N, the software must perform a corresponding number of instantiations sample-synchronously in addition to sound synthesis. On the Core i7-4500U, it can achieve 9 steps or up to 512 instantiations. Polyphony could be increased by staggering the instantiations in time, increasing latency (and thus amortization) or grouping several sinusoids in a single instance – using oscillator banks.

4.3 Score Auralization

The final example, in Listing 7 demonstrates a simplistic MUSIC-N descendant [2] system written entirely as a single Kronos program. The program defines three functions: a unit generator *(Exp-Gen)*, an instrument *(MyInstr)* and a score player/transformer *(MyPlayer)*. The score is defined as a matrix value *(MyScore)*.

Exp-Gen is the sole representative of Kronos' core capability: signal processing. It is an exponential function generator working at audio rate, consisting of a mutiplier and unit delay feedback. In this example it is used with complex-valued parameters, resulting a machine code procedure with just a handful of instructions per generated sample. *Exp-Gen* returns a reactive stream of floating point scalars.

MyInstr is an instrument wrapper for the *Exp-Gen* generator, receiving high level parameters for duration, pitch and amplitude. It computes complex coefficients based on them, instantiates the unit generator and schedules it to stop after the amplitude has decayed sufficiently. *MyInstr* returns an I/O action that performs these steps.

MyPlayer applies *MyInstr* to the notes in the score. The score is a list of 4-value tuples. The first value indicates the start time of a note, followed by the parameters required by *MyInstr*, duration, note number and amplitude. The start time is consumed by the player, used to schedule instrument invocations with the *After* command. The rest of the tuple is passed directly on to the instrument invocation as parameters.

The layers presented are intentionally simplistic. Different parametrizations and hierarchies can be devised for abstractions like multi-timbral scores, nested scores or real-time capable instruments. For example, with a suitable nested score format, *MyPlayer* could schedule instances of itself that in turn schedule sub-scores. Such flexibility is the result of the general purpose capability of the Meta-Sequencer: with a handful of I/O hooks, an interpreter and a closely integrated high performance JIT compiler, temporal recursion [4] is sufficient for a wide range of musical constructs.

Listing 7. Simple Ugen, Instrument and Score

```
Import Complex
Import Actions

; Unit generator: output an exponential function
; Can produce a sinusoid with complex-valued params
Exp-Gen(init coef) {
  state = z-1(init state * Audio:Signal(coef))
  Exp-Gen = state
}

; Instrument: configure, start and stop the ugen
MyInstr(dur pitch amp) {
  Use Actions
  Use Math

  ; compute complex coefficients
  fsr = Audio:Rate()
  ; angular frequency from note number
  w = Pi * 880 * Pow(2 (pitch - 69) / 12) / fsr
  ; radius; decay of 1/100 in 'dur' time
  r = Pow(0.01 1 / (dur * fsr))
  coef = Complex:Polar(w r)
  init = Complex:Cons(0 amp)

  ; instantiate an ugen and stop it after
  ; twice the duration (decay of 1/10000)
  MyInstr = Do(
    id <- Start({
      Complex:Real(Exp-Gen(init coef)) })
    After(duration * 2 Stop(id))
  )
}

; Score format: <time> <duration> <note-number> <amplitude>
MyScore =
 [(0 3 60 1 )
  (1 2 64 1 )
  (2 1 67 1 )
  (3 0.1 72 0.5)
  (3.1 0.1 71 0.4)
  (3.2 0.1 70 0.3)
  (3.3 0.1 69 0.2)
  (3.4 0.1 68 0.1)]

; Construct and schedule 'MyInstr' instance for each
; note in the score.
MyPlayer(score tempo-scale instr) {
  Use Actions
  MyPlayer = For(score (time params) =>
     After(time * tempo-scale instr(params)))
}
; Usage: MyPlayer(MyScore 1 MyInstr)
```

4.4 Compiler Stack and Real-Time Playback

The Meta-Sequencer is sample-accurate and audio synchronous; the implication is that sometimes, JIT compilation must interrupt the real-time audio thread. Even simple code takes time to travel through the full LLVM stack; at minimum, compile times are in the order of milliseconds, making it hard to sustain uninterrupted real time playback.

However, the Meta-Sequencer is capable of synthesizing a surprising range of algorithms in real time, allowing for a small delay in the initial response. This is because of type determinisim in the Kronos language [3]: the compiler output depends uniquely on the type of the closure being compiled. This allows memoization of compiled closures based on their type, effectively reducing compilation of already-known closures to a simple hash table lookup.

The implication is important for a concept of *type loops* in temporal recursion. The type of each closure if determined by its captures and arguments. For example, the type loop in Listing 6 is closed: each recursion is *type-invariant* in its captures and arguments. In such a case, no additional compilation is required once the *type loop* has been completed.

5. FUTURE WORK

The introduction of the I/O language and temporally recursive sequencer extend the reach of the Kronos programming language upwards from signal processing towards representations of music and scores. This study describes the fundamentals required for such an extension; much work remains in fulfilling the nascent potential.

Immediate technical concerns include the compilation performance, as discussed in Secton 4.4. An interesting enhancement to the system would be to analyze scheduled ASTs for closures that could be compiled anticipatively.

Core Kronos features dynamic as well as static compilation. However, the AST interpreter requires a runtime component that is so far absent from statically compiled binaries. It is viable to produce such a run time and produce dependency-free binaries from Meta-Sequencer programs with *closed type loops* (see Section 4.4).

The development of the I/O Action language and related usability aspects is also an interesting avenue for future work. Enhancing the Kronos core library towards score metaphors and any potential problems thus uncovered in the compiler design represent an important strategy of incremental improvement.

Graphical representation of imperative programs, as well as integration to GUI tools, including PWGL and ENP [20] remain compelling.

6. CONCLUSIONS

This study presented the Meta-Sequencer, an extension to the Kronos programming language [3]. The implemetation of an I/O Action Language, sourcing from the concepts in the Haskell [12] I/O Monad, was discussed. The implications for musical applications, especially with the addition of temporal recursion [4] were explained and demonstrated.

The study represents an attempt to extend a signal processing language, previously focused on unit generator and orchestra programming towards scores and musical abstractions. Kronos is an ideal platform for such a work, as it focuses on meta-programming, extensibility and domain languages.

Acknowledgments

Vesa Norilo's work has been supported by the Emil Aaltonen Foundation.

7. REFERENCES

[1] C. Roads, *the Computer Music Tutorial*. Cambridge: MIT Press, 1996.

[2] V. Lazzarini, "The Development of Computer Music Programming Systems," *Journal of New Music Research*, vol. 42, no. 1, pp. 97–110, Mar. 2013. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/09298215.2013.778890

[3] V. Norilo, "Kronos: A Declarative Metaprogramming Language for Digital Signal Processing," *Computer Music Journal*, vol. 39, no. 4, 2015.

[4] A. Sorensen and H. Gardner, "Programming With Time Cyber-physical programming with Impromptu," *Time*, vol. 45, pp. 822–834, 2010. [Online]. Available: http://doi.acm.org/10.1145/1869459.1869526

[5] M. Puckette and D. Zicarelli, *MAX - An Interactive Graphical Programming Environment*. Opcode Systems, 1990.

[6] M. Puckette, "Pure data: another integrated computer music environment," in *Proceedings of the 1996 International Computer Music Conference*, 1996, pp. 269–272.

[7] J. McCartney, "Rethinking the Computer Music Language: SuperCollider," *Computer Music Journal*, vol. 26, no. 4, pp. 61–68, 2002.

[8] G. Wang, P. R. Cook, and S. Salazar, "ChucK: A Strongly Timed Computer Music Language," *Computer Music Journal2*, vol. 39, no. 4, pp. 10–29, 2015.

[9] Y. Orlarey, D. Fober, and S. Letz, "Syntactical and semantical aspects of Faust," *Soft Computing*, vol. 8, no. 9, pp. 623–632, 2004.

[10] C. Strachey, "Fundamental Concepts in Programming Languages," *Higher-Order and Symbolic Computation*, vol. 13, no. 1-2, pp. 11–49, 2000.

[11] E. Brandt, "Temporal type constructors for computer music programming," Ph.D. dissertation, Carnegie Mellon University, 2002.

[12] P. Hudak, J. Hughes, S. P. Jones, and P. Wadler, "A history of Haskell," *Proceedings of the third ACM SIGPLAN conference on History of programming languages HOPL III*, pp. 12–1–12–55, 2007. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1238844.1238856

[13] P. Hudak, "Conception, evolution, and application of functional programming languages," *ACM Computing Surveys*, vol. 21, no. 3, pp. 359–411, 1989.

[14] M. Wright, A. Freed, and A. Momeni, "OpenSound Control: State of the Art 2003," in *Proceedings of NIME*, Montreal, 2003, pp. 153–159.

[15] V. Norilo, "Visualization of Signals and Algorithms in Kronos," in *Proceedings of the International Conference on Digital . . .* , York, 2012, pp. 15–18.

[16] ——, "Introducing Kronos - A Novel Approach to Signal Processing Languages," in *Proceedings of the Linux Audio Conference*, F. Neumann and V. Lazzarini, Eds. Maynooth: NUIM, 2011, pp. 9–16.

[17] R. B. Dannenberg, "Expressing Temporal Behavior Declaratively," in *CMU Computer Science, A 25th Anniversary Commemorative*, R. F. Rashid, Ed. ACM Press, 1991, pp. 47–68.

[18] G. Wakefield, W. Smith, and C. Roberts, "LuaAV: Extensibility and Heterogeneity for Audiovisual Computing," *Proceedings of the Linux Audio Conference*, 2010. [Online]. Available: https://mat.ucsb.edu/Publications/wakefield_smith_roberts LAC2010.pdf

[19] P. Van Roy, "Programming Paradigms for Dummies: What Every Programmer Should Know," in *New Computational Paradigms for Music*, G. Assayag and A. Gerzso, Eds. Paris: Delatour France, IRCAM, 2009, pp. 9–49.

[20] M. Laurson, M. Kuuskankare, and V. Norilo, "An Overview of PWGL, a Visual Programming Environment for Music," *Computer Music Journal*, vol. 33, no. 1, pp. 19–31, 2009.

Panoramix: 3D mixing and post-production workstation

Thibaut Carpentier
UMR 9912 STMS IRCAM – CNRS – UPMC
1, place Igor Stravinsky, 75004 Paris
thibaut.carpentier@ircam.fr

ABSTRACT

This paper presents panoramix, a post-production workstation for 3D-audio contents. This tool offers a comprehensive environment for mixing, reverberating, and spatializing sound materials from different microphone systems: surround microphone trees, spot microphones, ambient miking, Higher Order Ambisonics capture. Several 3D spatialization techniques (VBAP, HOA, binaural) can be combined and mixed simultaneously in different formats. Panoramix also provides conventional features of mixing engines (equalizer, compressor/expander, grouping parameters, routing of input/output signals, etc.), and it can be controlled entirely via the Open Sound Control protocol.

1. INTRODUCTION

Sound mixing is the art of combining multiple sonic elements in order to eventually produce a master tape that can be broadcast and archived. It is thus a crucial step in the workflow of audio content production. With the increasing use of spatialization technologies in multimedia creation and the emergence of 3D diffusion platforms (3D theaters, binaural radio-broadcast, etc.), new mixing and post-production tools become necessary.

In this regard, the post-production of an electroacoustic music concert represents an interesting case study as it involves various mixing techniques and raises many challenges. The mixing engineer usually has to deal with numerous and heterogeneous audio materials: main microphone recording, spot microphones, ambient miking, electronic tracks (spatialiazed or not), sound samples, impulse responses of the concert hall, etc. With all these elements at hand, the sound engineer has to reproduce (if not re-create) the spatial dimension of the piece. His/her objective is to faithfully render the original sound scene and to preserve the acoustical characteristics of the concert hall while offering a clear perspective on the musical form. Most often the mix is produced from the standpoint of the conductor as this position allows to apprehend the musical structure and provides an analytic point of view which conforms to the composer's idea.

Obviously, the sound recording made during the concert is of tremendous importance and it greatly influences the post-production work. Several miking approaches can be used (spaced pair, surround miking, close microphones, etc.), and advantages and drawbacks of each technique are well known (see for instance [1–4]). For instance when mixing Pierre Boulez's *Répons*, Lyzwa emphasized that multiple miking techniques had to be combined in order to benefit from their complimentarity [5]: a main microphone tree (e.g. surround 5.0 array) captures the overall spatial scene and provides a realistic impression of envelopment as the different microphone signals are uncorrelated; such a system is well suited for distant sounds and depth perception. However the localization of sound sources lacks precision, and thus additional spot microphones have to be used, close to the instruments. During post-production, these spot microphones have to be re-spatialized using panning techniques. Electronic tracks, if independently available, have to be processed similarly. Finally the sound engineer can add artificial reverberation in the mix in order to fuse the different materials and to enhance depth impression.

In summary, the mixing engineer's task is to create a comprehensive sound scene through manipulation of the spatial attributes (localization, immersion, envelopment, depth, etc.) of the available audio materials. Tools used in the post-production workflow typically consist of: a mixing console (analog or digital), digital audio workstations (DAWs) and sound spatialization software environments.

The work presented in this article aims at enhancing existing tools especially in regard to 3D mixing wherein existing technologies are ill-suited. Mixing desks are usually limited to conventional panning techniques (time or intensity differences) and they do not support 3D processing such as binaural or Ambisonic rendering. They are most often dedicated to 2D surround setups (5.1 or 7.1) and they do not provide knob for elevation control. Similarly, digital audio workstations lack flexibility for multichannel streams: most of the DAWs only support "limited" multichannel tracks/busses (stereo, 5.1 or 7.1) and inserting spatialization plugins is difficult and/or tedious. On the other hand, many powerful sound spatialization engines are available. As shown in [6] and other surveys, a majority of these tools are integrated into realtime media-programming environments such as Max or PureData. Such frameworks appear inadequate to post-production and mixing as many crucial operations (e.g. group management or dynamic creation of new tracks) can hardly be implemented. Furthermore, spatialization libraries are generally dedicated to one given rendering technique (for instance VBAP [7] or Higher-Order Ambisonic [8]) and they are ill-suited to hybrid mix.

Finally, high-spatial resolution microphones such as the EigenMike [1] are essentially used in research labs but they remain under-exploited in actual production contexts, in spite of their great potential.

As a consequence, we have developed a new tool which provides a unified framework for the mixing, spatialization and reverberation of heterogeneous sound sources in a 3D context.

This paper is organized as follows: Section 2 presents the process of recording an electroacoustic piece for use in 3D post-production. This paradigmatic example is used to elaborate the specifications of the new mixing engine. Section 3 details the technical features of *panoramix*, the proposed workstation. Finally Section 4 outlines possible future improvements.

[1] http://www.mhacoustics.com

2. PARADIGMATIC EXAMPLE

2.1 Presentation

Composer Olga Neuwirth's 2015 piece *Le* Encantadas *o le avventure nel mare delle meraviglie*, for ensemble and electronics [2] serves as a useful case study in 3D audio production techniques. The piece had its French premiere on October 21st in the Salle des Concerts de la Philharmonie 2 (Paris), performed by the Ensemble intercontemporain with Matthias Pintscher conducting. As is often the case in Neuwirth's work, the piece proposed a quite elaborate spatial design, with the ensemble divided in six groups of four or five musicians. Group I was positioned on-stage, while groups II to VI were dispatched in the balcony, surrounding and overlooking the audience (cf. Figure 1). The electronic part combined pre-recorded sound samples and real-time effects, to be rendered over a 40-speaker 3D dome above the audience. Different spatialization approaches were employed, notably Higher-Order Ambisonic (HOA), VBAP, and spatial matrixing. Throughout the piece, several virtual sound spaces were generated by means of reverberators. In particular, high-resolution directional room impulse responses, measured with an EigenMike microphone in the San Lorenzo Church (Venice), were used in a 4th order HOA convolution engine in order to simulate the acoustics of the church – as a reference to Luigi Nono's *Prometeo*.

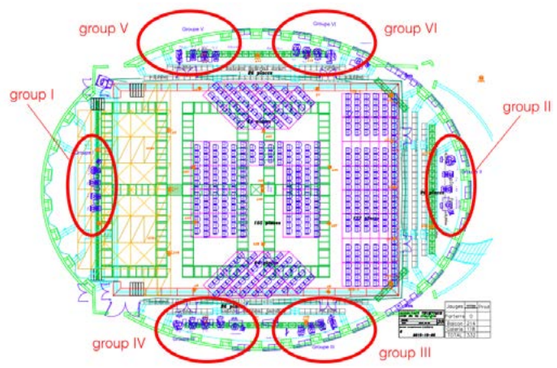


Figure 1. Location of the six instrumental groups in the Salle des Concerts – Philharmonie 2, Paris.

[2] Computer music design: Gilbert Nouno / Ircam

group	instruments	miking
I	saxophone, trumpet 1, bassoon, electric guitar	4 microphones: AT4050, AKG214, C535, AKG214
II	synthesizer 1, clarinet 1, trumpet 2, cello 1	5 microphones: KMS105, DPA4021, AKG214, KM140, AKG411
III	flute 1, oboe, french horn 1, trombone 1, percussion 1	11 microphones: DPA4066, KM150, C353, KM150, Beta57, SM58 (x2), SM57 (x2), C535, AKG411
IV	synthesizer 2, violin 3, violin 4, viola 1, cello 2	5 microphones: DPA4061 (x3), DPA2011, KM140
V	percussion 2, trombone 2, french horn 2, clarinet 2, flute 2	10 microphones: SM57 (x2), SM58 (x2), MD421, C535, Beta57, KMS105, AKG414 (x2), DPA4066
VI	synthesizer 3, violin 1, violin 2, viola 2, double bass	10 microphones: DPA4061 (x3), AKG414 (x4), KM140, C535 (x2), SM58 (x2)

Table 1. Spot microphones used for the recording.

2.2 Sound recording

Given the spatial configuration of the piece, the recording session [3] involved a rather large set of elements:

- 45 close microphones for the six instrumental groups (see Table 1),
- distant microphones for capturing the overall image of the groups: spaced microphones pairs for groups I and II; omni-directional mics for the side groups,
- one EigenMike microphone (32 channels) in the middle of the hall, i.e. in the center of the HOA dome,
- one custom 6-channel surround tree (see [5]) also located in the center of the hall,
- 32 tracks for the electronics (30 speaker feeds plus 2 subwoofers),
- direct capture of the 3 (stereo) synthesizers as well as 3 click tracks.

In total, 132 tracks were recorded with two laptop computers (64 and 68 channels respectively) which were later re-synchronized by utilizing click tracks.

2.3 Specifications for the post-production workstation

In spite of its rather large scale, this example of recording session is representative of what is commonly used in the electroacoustic field, where each recorded element requires post-production treatment. As mentioned in the introduction, various tools can be used to handle these treatments, however there is yet no unified framework covering all the required operations.

Based on the example of *Encantadas* (and others not covered in this article), we can begin to define the specifications for a comprehensive mixing environment. The workstation should (at least) allow for:

- spatializing monophonic sound sources (spot microphones or electronic tracks) in 3D,
- adding artificial reverberation,
- encoding and decoding of Ambisonic sound-fields (B-format or higher orders),
- mixing already spatialized electronic parts recorded as speaker feeds,

[3] Sound recording: Ircam / Clément Cornuau, Mélina Avenati, Sylvain Cadars

- adjusting the levels and delays of each elements so as to align them,
- combining different spatialization approaches,
- rendering and exporting the final mix in several formats.

With these specifications in mind, we developed *panoramix*, a virtual mixing console which consists of an audio engine associated with a graphical user interface for controlling/editing the session.

3. PANORAMIX

Like a traditional mixing desk, the *panoramix* interface is designed as vertical strips depicted in Figure 3. Strips can be of different types, serving different purposes with the following common set of features:

- multichannel vu-meter for monitoring the input level(s),
- input trim,
- multichannel equalization module (where the EQ is applied uniformly on each channel). The equalizer comes as a 8-stage parametric filter (see ⑥ in Figure 3) with one high-pass, one low-pass (Butterworth design with adjustable slope), two shelving filters, and four second-order sections (with adjustable gain, Q and cutoff frequency),
- multichannel dynamic compressor/expander (Figure 2) with standard parameters (ratio, activation threshold, and attack/release settings),
- mute/solo buttons,
- multichannel vu-meter for output monitoring, with a gain fader.

In addition, a toolbar below the strip header (Figure 3), allows for the configuration of various options such as locking/unlocking the strip, adding textual annotations, and configuring the vu-meters (pre/post fader, peakhold), etc. Strips are organized in two main categories: input tracks and busses. The following sections describe the properties of each kind of strip.

3.1 Input tracks

Input tracks correspond to the audio streams used in the mixing session (which could be real-time or prerecorded). Each input track contains a delay parameter in order to re-synchronize audio recorded with different microphone systems. For example, spot microphones are recorded close to the instruments and so their signals arrive faster than microphones placed at greater distances. Miking a sound source with multiple microphones is also prone to tone coloration; adjusting the delay parameter helps reducing this coloration and can also be used to vary the sense of spatial envelopment. In practice, it can be effective to set the spot microphones to arrive slightly early, to take advantage of the precedence effect which stabilizes the perceived location of the combined sound.

3.1.1 Mono Track

A Mono Track is used to process and spatialize a monophonic signal, typically from a spot microphone or an electronic track. The strip provides controls over the localization attributes (azimuth, elevation, distance), spatial effects (Doppler, air absorption filtering) and reverberation. The artificial reverberation module is derived from the Spat architecture [9] wherein the generated room effect is composed

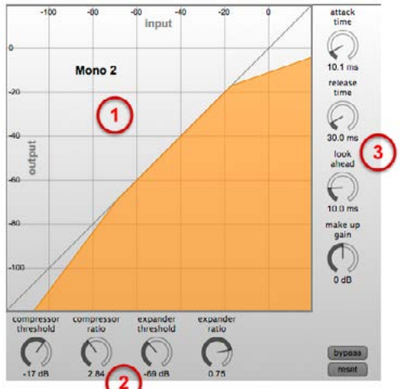


Figure 2. Compressor/expander module. ① Dynamic compression curve. ② Ratios and thresholds. ③ Temporal characteristics.

of four temporal sections: direct sound, early reflections, late/diffuse reflections and reverberation tail. By default the Spat perceptual model is applied, using the source distance to calculate the gain, delay, and filter coefficients for each of the four temporal sections. Alternatively, the perceptual model can be disabled (see slave buttons ③ in Figure 4) and the levels manually adjusted. Each temporal section may also be muted independently. In the signal processing chain, the extended direct sound (i.e. direct sound plus early reflections) is generated inside the mono track (Figure 7), while the late/diffuse sections are synthesized in a reverb bus (described in 3.2.2) which is shared among several tracks in order to minimize the CPU cost. Finally, a drop-down menu ("bus send") allows one to select the destination bus (see 3.2.1) of the track.

Moreover all mono tracks are visualized (and can be manipulated) in a 2D geometrical interface (⑦ in Figure 3).

3.1.2 Multi Track

A Multi Track is essentially a coordinated collection of mono tracks, where all processing settings (filters, reverberation, etc.) are applied similarly on each monophonic channel. The positions of each of the mono elements are fixed (i.e. they are set once –via the "Channels..." menu– for the lifetime of the session). Such Multi Track is typically used to process a multichannel stream of speaker feeds signals (see paragraph 2.3).

Similar results could be obtained by grouping (see 3.5) multiple "Mono" tracks, however "Multi" tracks make the configuration and management of the session much more simple, rapid and intuitive.

3.1.3 EigenMike Track

As its name suggests, an "EigenMike" Track is employed to process recordings made with spherical microphone arrays such as the EigenMike. Correspondingly, the track has 32 input channels and it encodes spherical microphone signals in the HOA format. Encoding can be performed up to 4^{th} order, and several normalization flavors (N3D, SN3D, FuMa, etc.) are available.

Modal-domain operators can later be applied to spatially transform the encoded sound-field, for example rotating the

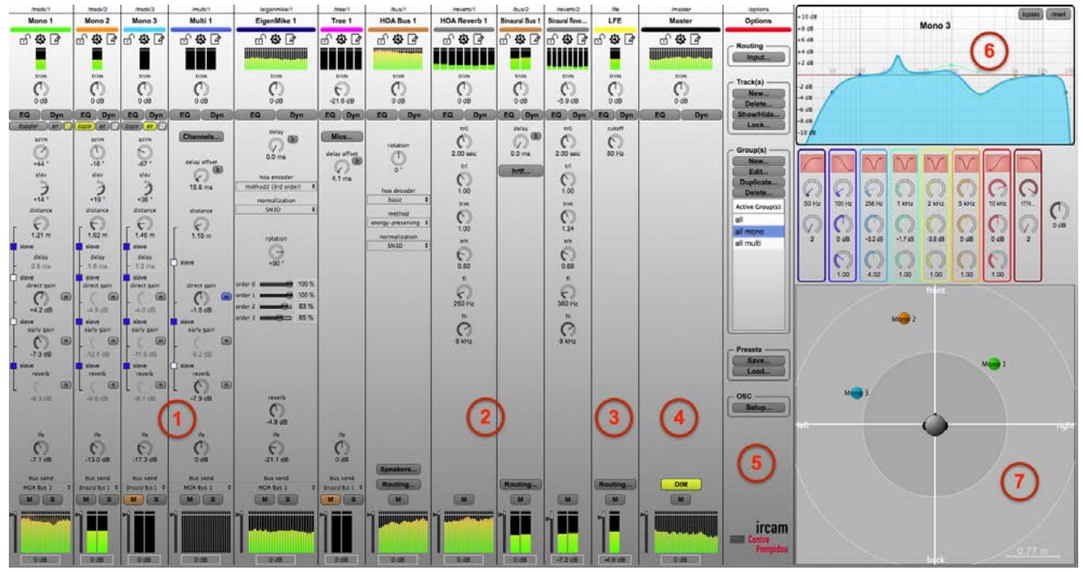


Figure 3. Main interface of the *panoramix* workstation. ① Input strips. ② Panning and reverb busses. ③ LFE bus. ④ Master track. ⑤ Session options. ⑥ Insert modules (equalizer, compressor, etc.). ⑦ Geometrical interface for positioning.

whole sound scene, or weighting the spherical harmonics components (see ④ in Figure 4).

Signals emanating from an EigenMike recording are already spatialized and they convey the reverberation of the concert hall, however a reverb send parameter is provided in the track, which can be useful for adding subtle artificial reverberation, coherent with the other tracks, to homogenize the mix. The reverb send is derived from the omni component (W-channel) of the HOA stream.

3.1.4 Tree Track

A "Tree" track is used to accommodate the signals of a microphone tree such as the 6-channel tree installed for the recording of *Encantadas* (section 2.2). The "Mics..." button (cf. Track "Tree 1" in Figure 3) pops up a window for setting the positions of the microphones in the tree. It is further possible to align the delay and level of each cell of the microphone array.

As microphone trees entirely capture the sound scene, the "Tree" track does not apply any specific treatment to the signals.

3.2 Busses

Three types of bus are provided: panning busses, reverb busses, and one LFE ("low frequency enhancement") bus.

3.2.1 Panning/Decoding bus

The role of panning busses is threefold: 1) they act as summing busses for the track output streams; 2) they control the spatialization technique in use (three algorithms are currently supported: VBAP, HOA and binaural); 3) panning busses are used to control various parameters related to the encoding/decoding of the signals. For speaker-based rendering (VBAP or HOA), the "Speakers..." button allows for the configuration of the speakers layout (Figure 6); in case of binaural reproduction, the "hrtf..." button provides means to select the desired HRTF set. Finally, HOA panning busses decode the Ambisonic streams, and several decoding parameters can be adjusted (see "HOA Bus 1" in Figure 3).

The selection of rendering techniques (VBAP, HOA, binaural) was motivated by their ability to spatialize sounds in full 3D, and their perceptual complementarity. Other panning algorithms may also be added in future versions of *panoramix*.

Output signals from the panning busses are sent to the Master strip. Each panning bus provides a routing matrix so as to assign the signals to the desired destination channel (② in Figure 5).

3.2.2 Reverberation bus

Reverberation busses function to synthesize the late/diffuse sections of the artificial reverberation processing chain. A reverb bus is uniquely and permanently attached to one or more panning buses, where the reverberation effect is applied to each track routed to this bus.

Panoramix builds on the reverberation engine of Spat which consists of a feedback delay network with an variable decay profile, adjustable in three frequency bands. The main parameters of the algorithm are exposed in the reverb strip (see ⑤ in Figure 4).

3.2.3 LFE Bus

Each track has a LFE knob to tune the amount of signals sent to the LFE bus which handles the low-frequency signals sent to the subwoofer(s) of the reproduction setup. The bus applies a low-pass filter with adjustable cutoff frequency.

3.3 Master

The "Master" strip collects the output signals of all the busses and forwards them to the *panoramix* physical outputs.

Although the workstation only has one Master strip, it is possible to simultaneously render mixes in various formats. For instance, if the session has 26 physical output channels, one can assign channels 1–24 to an Ambisonic mix and channels 25–26 to a binaural rendering.

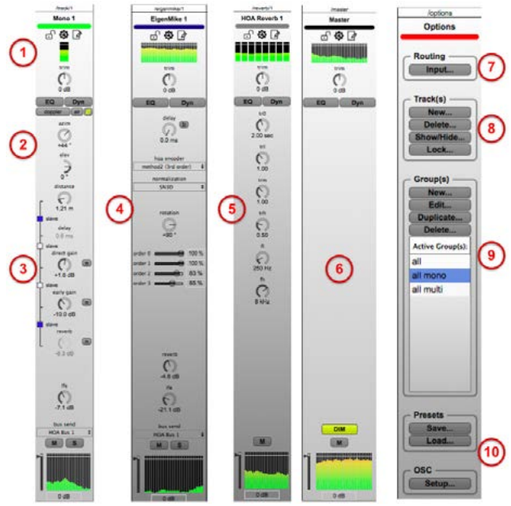


Figure 4. View of multiple strips; from left to right: mono track, EigenMike track, HOA reverberation bus, master track, session options. ① Strip header: name of the strip, color, lock/unlock, options, annotations, input vu-meter, input trim, equalizer and compressor. ② Localization parameters (position, Doppler effect, air absorption). ③ Room effect settings (direct sound, early reflections, send to late reverb). ④ HOA encoding and sound-field transformations parameters. ⑤ Late reverb settings (reverberation time, crossover frequencies, etc.). ⑥ Master track. ⑦ Input matrix. ⑧ Track management (create, delete, etc.). ⑨ Groups management. ⑩ Import/export of presets and OSC configuration.

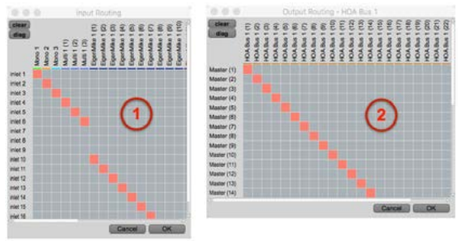


Figure 5. ① Input routing. Physical inputs (rows of the matrix) can be assigned to the available tracks (columns). ② Panning bus routing "HOA 1". The output of the bus (columns) can be routed to the Master channels (rows), i.e. towards the physical outputs.
Each channel can have multiple connections (e.g. one physical input can be routed to several tracks).

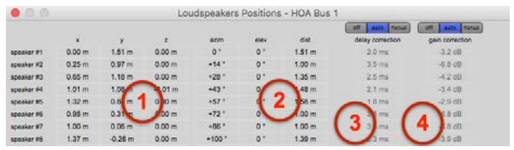


Figure 6. Configuration of the speaker layout for a panning bus. Speakers coordinates can be edited in Cartesian ① or spherical ② coordinates. The reproduction setup can be aligned in time ③ and level ④; delays and gains are automatically computed or manually entered.

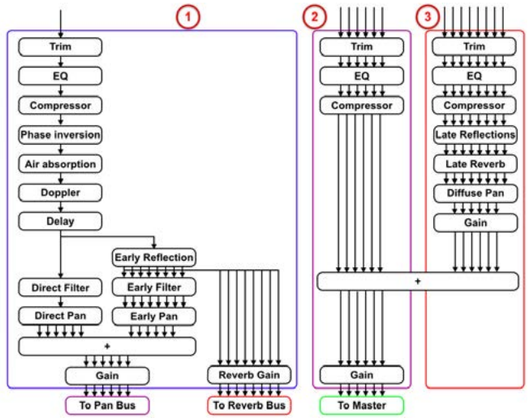


Figure 7. Audio architecture (simplified representation). ① Mono track. ② Panning/decoding bus. ③ Reverb bus.

3.4 Session options

The "Options" strip is used for the management of the mixing session. This includes routing of the physical inputs (see ⑦ in Figure 4 and ① in Figure 5), creation and edition of the tracks and busses (⑧ in Figure 4) as well as import/export of preset files (⑩ in Figure 4).

3.5 Group management

In a mixing context, it is frequently useful to group (or link) several parameters to maintain a coherent relationship while manipulating them. To achieve this, *Panoramix* offers a grouping mechanism where all modifications to one track parameter will also offset that parameter in every linked track. The "Options" strip provides a means to create, edit, duplicate or delete groups (see ⑨ in Figure 4 and Figure 8), and the ability to select the active group(s). Grouping effects all track parameters by default, however it is also possible to exclude some parameters from the group (e.g. mute, solo, send; see ③ in Figure 8).

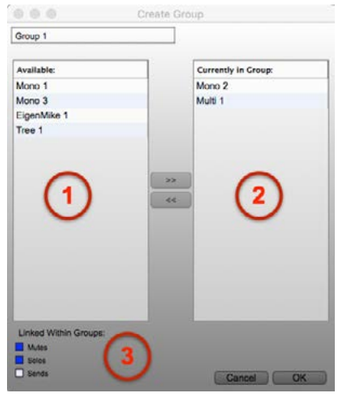


Figure 8. Creation/edition of a group. ① Available tracks. ② Tracks currently in group. ③ Group options.

3.6 OSC communication

All parameters of the *panoramix* application can be remotely accessed via the Open Sound Control protocol (OSC [10]). Typically, a digital audio workstation is used for edition and playback of the audio tracks while *panoramix* handles the spatial rendering and mixing (see Figure 9). Automation data is stored in the DAW and sent to *panoramix* through OSC via a plugin such as ToscA [11].

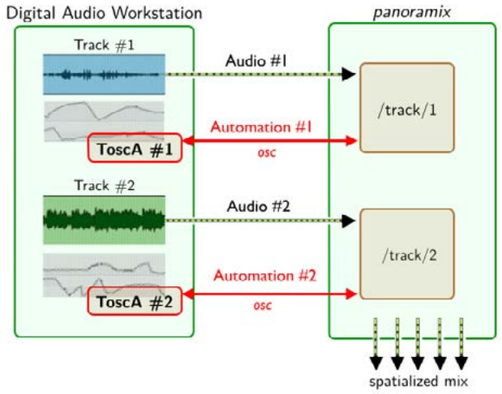


Figure 9. Workflow with *panoramix* and a digital audio workstation communicating through the OSC protocol and the ToscA plugin.

4. CONCLUSION AND PERSPECTIVES

This paper considered the design and implementation of a 3D mixing and post-production workstation. The developed application is versatile and offers a unified framework for mixing, spatializing and reverberating sound materials from different microphone systems. It overcomes the limitations of other existing tools and has been proved useful in practical mixing situations.

Nonetheless, the application can be further improved and many new features are considered for future versions. This includes (but is not limited to):

- support of other encoding/decoding strategies, notably for M-S and B-Format microphones,
- extension of the reverberation engine to convolution or hybrid processors [12],
- import and/or export of the tracks' settings in an object-oriented format such as ADM [13],
- implementation of monitoring or automatic down-mixing tools, based for instance on crosstalk cancellation techniques as proposed in [14],
- insert of audio plugins (VST, AU, etc.) in the strips,
- integration of automation data directly into the *panoramix* workstation,
- synchronization of the session to a LTC time-code.

Acknowledgments

The author is very grateful to Clément Cornuau, Olivier Warusfel, Markus Noisternig and the whole sound engineering team at Ircam for their invaluable help in the conception of this tool. The author also wish to thank Angelo Farina for providing the EigenMike used for the recording of *Encantadas*, and Olga Neuwirth for authorizing this recording and its exploitation during the mixing sessions.

5. REFERENCES

[1] D. M. Huber and R. E. Runstein, *Modern Recording Techniques (8th Edition)*. Focal Press, 2014.

[2] F. Rumsey and T. McCormick, *Sound and Recording (6th edition)*. Elsevier, 2009.

[3] B. Bartlett, "Choosing the Right Microphone by Understanding Design Tradeoffs," *Journal of the Audio Engineering Society*, vol. 35, no. 11, pp. 924 – 943, Nov 1987.

[4] R. Knoppow, "A Bibliography of the Relevant Literature on the Subject of Microphones," *Journal of the Audio Engineering Society*, vol. 33, no. 7/8, pp. 557 – 561, July/August 1985.

[5] J.-M. Lyzwa, "Prise de son et restitution multicanal en 5.1. Problématique d'une œuvre spatialisée : Répons, Pierre Boulez," Conservatoire National Supérieur de Musique et de Danse de Paris, Tech. Rep., May 2005.

[6] N. Peters, G. Marentakis, and S. McAdams, "Current Technologies and Compositional Practices for Spatialization: A Qualitative and Quantitative Analysis," *Computer Music Journal*, vol. 35, no. 1, pp. 10 – 27, 2011.

[7] V. Pulkki, "Virtual Sound Source Positioning Using Vector Base Amplitude Panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456 – 466, June 1997.

[8] J. Daniel, "Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia," Ph.D. dissertation, Université de Paris VI, 2001.

[9] T. Carpentier, M. Noisternig, and O. Warusfel, "Twenty Years of Ircam Spat: Looking Back, Looking Forward," in *Proc. of the 41st International Computer Music Conference*, Denton, TX, USA, Sept. 2015, pp. 270 – 277.

[10] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 3, pp. 193 – 200, Dec 2005.

[11] T. Carpentier, "ToscA: An OSC Communication Plugin for Object-Oriented Spatialization Authoring," in *Proc. of the 41st International Computer Music Conference*, Denton, TX, USA, Sept. 2015, pp. 368 – 371.

[12] T. Carpentier, M. Noisternig, and O. Warusfel, "Hybrid Reverberation Processor with Perceptual Control," in *Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx-14)*, Erlangen, Germany, Sept. 2014.

[13] M. Parmentier, "Audio Definition (Metadata) Model – EBU Tech 3364," European Broadcasting Union, Tech. Rep., 2015. [Online]. Available: https://tech.ebu.ch/docs/tech/tech3364.pdf

[14] A. Baskind, T. Carpentier, J.-M. Lyzwa, and O. Warusfel, "Surround and 3D-Audio Production on Two-Channel and 2D-Multichannel Loudspeaker Setups," in *3rd International Conference on Spatial Audio (ICSA)*, Graz, Austria, Sept. 2015.

Multi-Point Nonlinear Spatial Distribution of Effects across the Soundfield

Stuart James
Edith Cowan University
s.james@ecu.edu.au

ABSTRACT

This paper outlines a method of applying non-linear processing and effects to multi-point spatial distributions of sound spectra. The technique is based on previous research by the author on non-linear spatial distributions of spectra, that is, timbre spatialisation in the frequency domain. One of the primary applications here is the further elaboration of timbre spatialisation in the frequency domain to account for distance cues incorporating loudness attenuation, reverb, and filtration. Further to this, the same approach may also give rise to more non-linear distributions of processing and effects across multi-point spatial distributions such as audio distortions and harmonic exciters, delays, and other such parallel processes used within a spatial context.

1. INTRODUCTION

Controlling large multi-parameter systems has always been bound by evaluating on one side the extremities of performer specificity versus generality on the other. Is it possible to intentionally control thousands of parameters simultaneously in performance, particularly when each parameter may require an assortment of attributes such as source localization, source distance, source width, loudness, and frequency? Certainly traditional approaches to live performance using a standard mixing console present difficulties when diffusing multiple sound sources across a multi-loudspeaker system. As Jonty Harrison (2005) has stated on this issue:

> If you've got an eight-channel source, and every channel of the eight has a fader, how do you do crossfades? You haven't got enough hands! (Mooney (Ed.), 2005, Appendix 2) [1]

The author proposed a solution that involved mapping audio signals to some audio-rate multi-channel panning routines developed by the author.[1] The use of audio signals for control allowed for both synchrony and adequate timing resolution, without necessarily compromising data precision. Three audio signals were used to determine the spatial localization cues azimuth, distance, and elevation/zenith. These often comprised of a vector of Cartesian (x, y, z) coordinates. In order to control the state of independent spectra, these audio signals are de-interleaved. For example, to control 1024 spectral bands independently, 1024 parameter values are de-interleaved every 1024 audio samples [2].

The author also extended this to include a table lookup stage that would be used to determine how frequencies are distributed across space. In this way, a graphics file or video could be used to control this distribution in real-time. This novel process was described by the author as using *Wave Terrain Synthesis* as a framework for controlling another process, in this case timbre spatialisation in the frequency domain [3, 4].

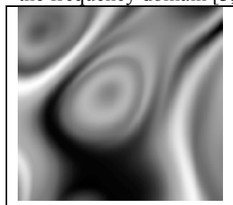


Figure 1a. A greyscale contour plot of a non-linear 2D table. Differences in colour are mapped to differences in frequency.

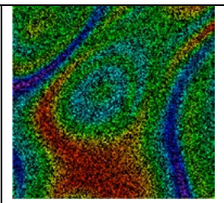


Figure 1b. A birds-eye view representing the spatial distribution of frequencies over 1 second using an asynchronous 2D random audio signal looking up values from the image in Figure 1a.

Schumacher and Bresson (2010) use the term 'spatial sound synthesis' to denote any sound synthesis process that is extended to the spatial domain [5]. Whilst timbre spatialisation [4, 10] falls into this category, other techniques include spatial swarm granulation [6], sinusoidal modulation synthesis [7], spectral spatialisation [8, 9], and spatio-operational spectral (SOS) synthesis [11].

2. TIMBRE SPATIALISATION IN THE FREQUENCY DOMAIN

The use of *Wave Terrain Synthesis* for controlling such a system relies on both the state of a stationary or evolving audio-rate *trajectory*, and the stationary or evolving state of a haptic-rate *terrain*. In this section some of these combinations of terrain and trajectory types are discussed in practice before extending the process to explore the impression of distance cues and other increasingly more non-linear approaches to spatial effects. Generally the results fall into the immersive category, but results can also be quite localised too.

[1] The author implemented audio-rate models of both Ambisonic Equivalent Panning (AEP) and Distance-based Ampliude Panning (DBAP)

For a single stationary *trajectory* over a colored terrain surface (a density plot using the color spectrum to describe the contour) only a single band of frequency is produced in the relative position of the virtual stationary point as shown in Figure 2a. Figure 2b shows the spectral processing functions (SPFs) that are produced for the four loudspeakers. These are color coded to illustrate the spectral distribution for each speaker. Since the point is closest to speaker 4 in Figure 2a, most of the energy accumulates in one speaker as shown in Figure 2b. In this case the amplitude ratio of this frequency is over 30 times, correlating with an increase in level of approximately +30 dB.

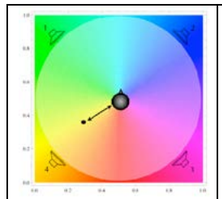


Figure 2a. A trajectory at a constant position of (0.35, 0.35).

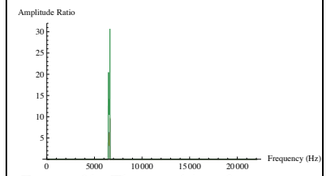


Figure 2b. The resulting frequency–amplitude for four speakers that accumulate over one frequency band.

For a circular *trajectory* across the listener field, synchronized to the frequency of the FFT, and such that the radius is equidistant about the virtual central (ideal) listening position, generates an even spread of frequencies around the listener as shown in Figure 3b. We notice here that there are four bands of frequency separated by the speakers with which they coincide. The panning algorithm ultimately determines the relative amplitude weighting of components across the speaker array. After the smoothing process (spectral centroid smoothing and linear-phase filtration) the frequency bands shift in level to a generalised weighting of four or an increase of +12 dB. Since this difference is substantial, the smoothing algorithms adopt an auto-normalise option that recalibrates automatically for large level differences introduced by the spatialisation process. This is calculated based on the relative loudness of the input source to be spatialised, and the resulting output level of the multi-channel audio.

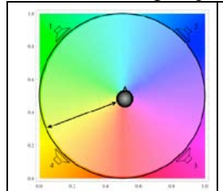


Figure 3a. A circular trajectory passing over a terrain where frequencies (shown in grey-scale) are distributed spatially.

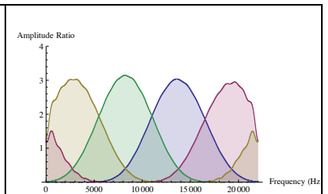


Figure 3b. The SPFs in Figure 3a after spectral centroid smoothing and linear-phase filtration have been applied.

A very unique outcome arises when the *terrain* and the *trajectory* curve are symmetrical about the vertical or horizontal axes, resulting in the same SPF being produced in multiple speakers. Any asymmetry in either the terrain or trajectory will result in different SPF functions for all speakers. Figure 4 shows a scenario where the SPFs for all speakers are different, yet still exhibit some symmetrical relationships.

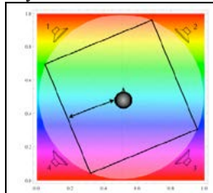


Figure 4a. A vertically symmetrical terrain curve, with a vertically and horizontally asymmetrical trajectory, and a vertically and horizontally symmetrical speaker configuration.

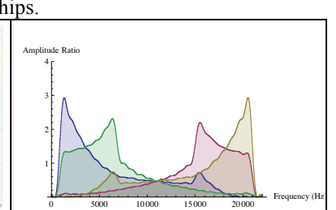


Figure 4b. The frequency–amplitude curves for all four speakers after spectral centroid smoothing and linear-phase filtration have been applied.

This scenario does not apply to terrain surfaces and/or trajectories that are not symmetrical over the horizontal or vertical axes. Sound shapes generated by non-symmetrical relationships result in all speakers having vastly different timbres as shown in Figure 5.

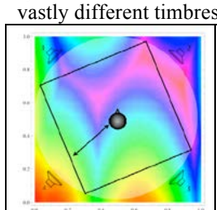


Figure 5a. An asymmetrical and non-linear terrain curve, with a vertically and horizontally asymmetrical trajectory, and a vertically and horizontally symmetrical speaker configuration.

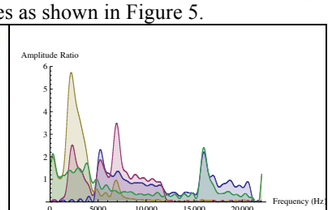


Figure 5b. The frequency–amplitude curves of the terrain and trajectory in Figure 6a through *Model B* showing a different spectrum in all four speakers. These spectral processing functions have had spectral centroid smoothing and linear-phase filtration applied.

Noisier signals increase the potential for describing a sound shape in more detail due to their more effective space-filling properties. Figure 6 shows a high-frequency asymmetrical trajectory used over a non-linear and asymmetrical terrain curve, resulting in a much more detailed series of SPFs generated.

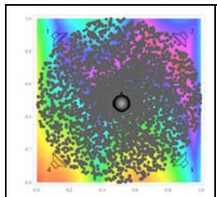


Figure 6a. A noisy high-frequency asynchronous trajectory passed over a non-linear terrain curve.

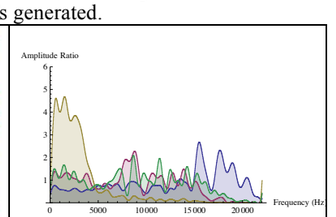


Figure 6b. The frequency–amplitude curves of the terrain and trajectory in Figure 6a. These spectral processing functions have had spectral centroid smoothing and linear-phase filtration applied.

The spatial resolution of these sound shapes can increase drastically with larger numbers of loudspeakers. In Figure 7, we see the same contour distributed between 1, 2, 8 and 32 speakers. The higher the number of loud-

speakers, the more spatial resolution, hence the spectral bands become increasingly separated. This enables the frequency response curves to represent the states 'in between'. As the number of speakers increases we observe increasing detail in each subsequent area of the spatial field determined by their respective set of SPF functions.

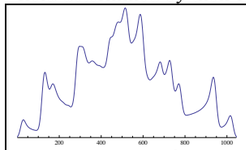


Figure 7a. A frequency–amplitude curve applied to one loudspeaker.

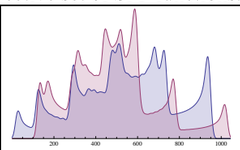


Figure 7b. The same frequency–amplitude curve applied to two loudspeakers.

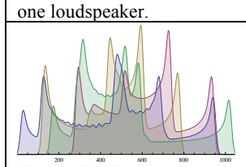


Figure 7c. The same frequency–amplitude curve applied to eight loudspeakers.

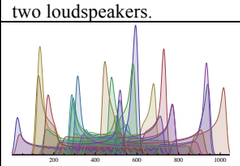


Figure 7d. The same frequency–amplitude curve applied to 32 loudspeakers.

3. DISTANCE CUES

One of the further lines of inquiry that emerged from this research involved integrating distance cues into such a model. What is commonly referred to as 'localisation' research is often only concerned with the direction of a source, whereas the perceived location of a sound source in a natural environment has two relatively independent dimensions—both direction and distance [12]. *Interaural intensity differences* (IIDs), *interaural time difference* (ITDs), and *spectral cues*, are significant in establishing a source sound's direction, but they do not take into consideration the perception of distance.[2] The perception of distance has been attributed to the loudness, the direct *v*. reflection ratio of a sound source, sound spectrum or frequency response due to the effects of air absorption, the initial time delay gap (ITDG), and movement [13].

Most software implementations that simulate direction and distance cues do not take into consideration the wide number of indicators for perceiving distance, as the algorithms responsible for panning sources (generally) only take into consideration differences in loudness; that is, they are often simply matrix mixers that control the various weights, or relative loudness, assigned to different speakers. However there is a small number of software implementations designed to additionally incorporate some of these other indicators for distance perception. These include implementations like *ViMiC* [14], *Spatialisateur* [15], and *OMPrisma* [16]. For example, *OMPrisma*, by Marlon Schumacher and Jean Bresson [17], includes pre-processing modules to increase the impression of distance and motion of a sound source. The effect of air absorption is accounted for using a second-order Butterworth low-pass filter, Doppler effects are simulated using a moving write-head delay line, and the decrease in amplitude (as a function of distance) is accomplished with a simple gain-stage unit.

3.1 Spatial Width

In addition to the spatial localization cues azimuth, distance, and elevation/zenith, the panning algorithms used in this research also included a further parameter determining the spatial width of each spectral bin. Spatial width[3] is considered to be another significant perceptible spatial attribute, and is defined as the perceived spatial dimension or size of the sound source [19]. The spatial width of sound sources is a natural phenomenon; for example, a beach front, wind blowing in trees, a waterfall and so on. Spatial width was incorporated in the model after observing the same approach used in implementations of Ambisonic Equivalent Panning, such as ICST[4] ambisonic panners for MaxMSP [18]. It should be made clear that Ambisonics algorithms do not render distance cues, however documentation by Neukom and Schacher [20] and its implementation in the ICST Ambisonics library demonstrate how the algorithm has been extended to account for distance. One of these relationships is the binding of spatial width with the distance of a sound source. The ICST implementation binds the order of directivity to the distance of each point, so as sources move further away from the centre they become narrower, and when they move closer they are rendered with greater spatial width, and if they are panned centre they are omnipresent. This is all dependent on the order of directivity of the AEP algorithm, as shown in Figure 8. Applying this at audio-rates with a polyphonic parameter system, like spectral processing, creates a complex spatial soundfield where different spectral bands have different orders of directivity.

Similarly other panning techniques such as Distance-based Amplitude Panning (DBAP) have provision for the amount of spatial blurring, which inadvertently increases the immersive effect, effectively spreading localized point-source movements to zones or regions of a multi-speaker array. Again, each spectral band can be rendered with a different spatial blur, resulting in a complex multi-parameter organization.

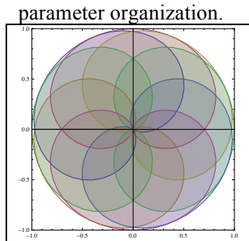


Figure 8a. Ambisonic equivalent panning (AEP) Order 1.

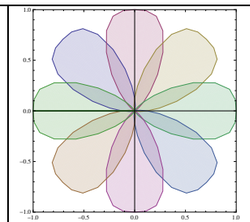


Figure 8b. AEP Order 16.

Whilst this could be determined solely by the radial distances of the intended diffusion, a further lookup stage could be used to determine spatial width across a 2D plane, either by a conventional circular distribution as shown in Figure 9, or one that is significantly more non-linear.

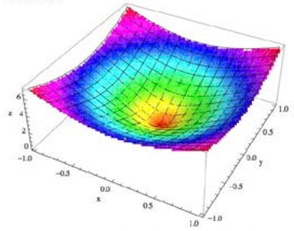


Figure 9. A circular distribution determining the order of directivity for different spatial coordinates (x, y).

3.2 Loudness

The role of loudness with respect to the perception of distance is inextricably linked with a sound sources relative drop in energy over distances, measured in decibels per metre (dB/m). The inverse distance law states that sound pressure (amplitude) falls inversely proportional to the distance from the sound source [24]. Distant sound sources have a lower loudness than close ones. This aspect can be evaluated especially easily for sound sources with which the listener is already familiar. It has also been found that closely moving sound sources create a different *interaural intensity difference* (ILD) in the ears than more distance sources [13].

However before considering the relative amplitudes generated across the multichannel system, we have to consider the amplitudes generated for each loudspeaker, keeping in mind the non-linearities of the panning algorithms used. For example, a complicating factor for the AEP model is that when incorporating more loudspeakers, and also modulation of the order of directivity, the resulting amplitude ranges change drastically too. Therefore implementations such as ICST account for both centre attenuation (dB) and distance attenuation (dB) (as well as the centre size). Centre attenuation is required to counteract the order of directivity when it is 0.[5] The distance attenuation serves to ensure that for larger virtual distances, the appropriate roll-off is applied. Some distance attenuation curves, with their associated parameter settings, are shown in Figure 10.

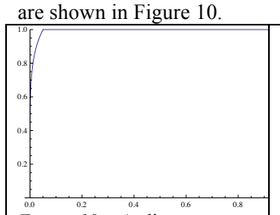


Figure 10a. A distance curve determined by centre size 0.05, centre attenuation 3 dB and distance attenuation 0 dB.

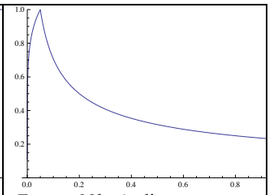


Figure 10b. A distance curve determined by centre size 0.05, centre attenuation 3 dB and distance attenuation 0.5 dB.

The frequency–amplitude curves generated in some cases can feature strong energy on certain bands of frequency, and this ultimately depends on the rate of change of the trajectory curve. In other words, stationary points in the *terrain* or *trajectory* are the reason for this accumulation of energy in certain regions of the frequency spectrum (see Figure 11a). Calibrating appropriate loudness attenuation curves across this 2D (or 3D system in the case of elevated cues) depend on relatively linear distributions of frequency across space. In order to achieve this, tests involved the use of a flat linear terrain surfaces, and a 2D random audio-rate trajectory with effective space-filling properties. Calibration of the distance as applied to *timbre spatialisation* can be achieved using the combination of a white noise *trajectory* over a simple linear *terrain* function. Figure 11b shows the standard frequency–space visualisation used in the authors research, and the ideal position of a listener (centre), where the distance of low frequencies highlighted (above) and low frequencies (below) are more distant than the mid-range frequencies (in the middle) that should sound perceptively louder.

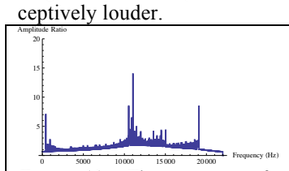


Figure 11a. The spectrum of a sound shape derived by the rose curve used as a trajectory over a linear ramp function. The rose curve features three stationary points.

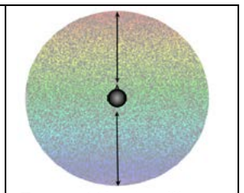


Figure 11b. An illustration explicitly pointing out that more distant frequencies in relation to the listener position need to be rolled off in loudness.

By reading the resulting frequency–amplitude curves from this process, it is possible to determine to what extent frequencies that are further away from the centre position are attenuated as a result of their relative distance from the listener, as shown in Figure 12a. These frequency–amplitude curves can be used to calibrate the distance roll-off curve and centre size of AEP. The combined use of the centroid smoothing and a linear-phase low-pass filter can also help to smooth out the peaks in the SPF in order to better gauge the roll-off in each instance. These smoothed frequency–amplitude plots are shown in Figures 12b. With a centre size of one and a roll-off of 3 dB, the impression of distance is subtle but evident. The use of the low-pass filter can also remove the comb filtering effects of the SPFs that result from computing the histogram.

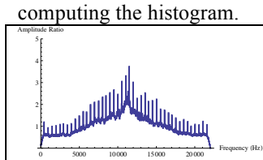


Figure 12a. A frequency–amplitude plot over 10 FFT frames with AEP centre size 0.05, centre attenuation 0 dB and distance attenuation 3 dB

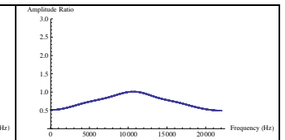


Figure 12b. The frequency–amplitude plot in Figure 115b with a linear-phase spectral low-pass filter applied.

As is the case with encoding spatial width, a 2D or 3D table can be used to lookup the relative loudness (or amplitude scaling) over a nominal distance.

[2] The attributes that assist in the perception of distance are sometimes referred to as *distance quality*.

[3] Also referred to in psychoacoustic literature as *spatial extent*, *source width* or *tonal volume*.

[4] The Institute for Computer Music and Sound Technology in Zürich, Switzerland.

[5] When the order of directivity is 0, the amplitude is 1 in all loudspeakers. Therefore for larger loudspeaker systems this accumulates based on number of speakers used.

3.3 Air Absorption

The sound spectrum can also be an indicator of distance since high frequencies are more quickly damped by air than low frequencies. Consequently, a distant sound source sounds more muffled than a close one, due to the attenuation of high frequencies. For sound with a known and limited spectrum—for example, human speech—the distance can be estimated roughly with the listener's prior knowledge of the perceived sound [25]. The implementation here effectively involves a parallel process that would essentially split the spectral bands based on a distance ratio. This involves an amplitude scaling function that is applied as the SPF functions are generated for each respective loudspeaker. By separating the spectra into two groups, one can be left a group of spectra that are unaffected (dry), whilst the other group is processed in some way (wet). In the case of air absorption, this would involve convolution filtering of the parallel group in order to attenuate high frequencies. As a result of this, perceptively the processing would appear to be applied increasingly more for distant spectra.

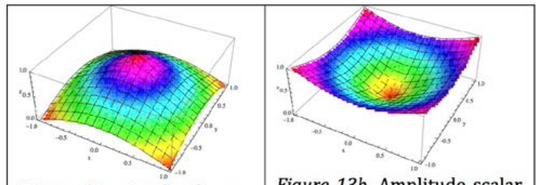


Figure 13a. Amplitude scalar for unprocessed spectra

Figure 13b. Amplitude scalar for processed spectra

3.4 Direct versus Reflection Ratio

The direct *v.* reflection ratio is a phenomenon that applies mostly to enclosed rooms and spaces. Typically two types of sound arrive at a listener: the direct sound source and the reflected sound. Reflected sound is sound that has been reflected at least once at a wall before arriving at the listener. In this way the ratio between direct sound and reflected sound can be an indicator of the distance of the sound source [13].

A way to integrate reverberation in such a multi-point model could be achieved in a similar way to the application of convolution filtration for simulating the effects of air absorption over larger distances. By separating the spectra into two groups, a dry and wet multi-point set, it is possibly to apply reverberation proportionally to the distant of each point of sound spectra from the central listening position. The amount of reverberation applied is therefore dependent on the distance quality of each frequency band.

The reverberation used may also allow for some adjustments in terms of the ratio of early reflection versus reverb tale, as well as the amount of pre-delay applied to the early reflections. If the pre-delay is short it may be indicative of a more distant sound source, versus a longer pre-delay indicating a first reflection that is heard off a nearby wall. This is often referred to as the Initial Time Delay Gap (ITDG). The ITDG describes the time difference between the arrival of the direct sound and first strong reflection at the listener. Nearby sound sources create a relatively large ITDG, with the first reflections having a longer path to the listener. When the source is far away, the direct and reflected sound waves have more similar path lengths.

The ITDG can be compensated for with the use of spectral delays, such that more distant frequency bands will be subjected to a different ITDG than a frequency band that is, in a virtual sense, closer to the listener. This aspect adds considerably more awareness of depth in the resulting spatialisation.

4. NONLINEAR SPATIAL DISTRIBUTION OF AUDIO EFFECTS

Another outcome of this same parallel process is firstly they could be used to apply other kinds of effects to a multi-point spatial distribution, and secondly they don't have to follow a distribution that is dependent on a central listener position, but rather aimed at exploring immersive and evolving transitions of effects such as delays, distortions, harmonic exciters, over a soundfield.

The fundamental process is the same here, where a spectral distribution is separated into an unprocessed group and a processed group. Figure 14 shows some nonlinear ways in which such a parallel process could manifest over a complex spatial sound shape.

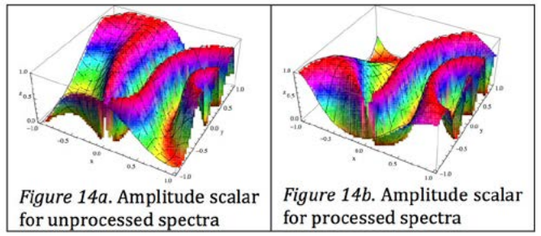


Figure 14a. Amplitude scalar for unprocessed spectra

Figure 14b. Amplitude scalar for processed spectra

5. CONCLUSIONS

Exploration of techniques that evoke a stronger sensation of distance in multi-point spatialisation, such as timbre spatialisation in the frequency domain, have resulted in more engaging spatial sound shapes with a stronger sense of depth over the soundfield. By applying some of these processes in parallel, it was also found that the same approach could be used to control other signal processes that are not specifically distance-dependent, but follow some other more novel and non-linear distribution across the soundfield. Further research could be focused on the movement of sound sources, particularly the effect known as 'Doppler shift'. The source *radial velocity*—the speed of a sound source moving through space—will affect the pitch of the sound due to the compression or expansion of the sound's wavelength as it travels through the air towards the listener [22]. Such effects may be possible through frequency modulating specific partials through the use of specific all-pass filters [23]. Furthermore, blindfold listener evaluation of such effects are essential in both evaluating the effectiveness, and optimizing the perceived effect of such processes.

6. REFERENCES

[1] Mooney, J. (Ed.) (2005). An Interview with Professor Jonty Harrison, In J. Mooney *Sound Diffusion Systems for the Live Performance of Electroacoustic Music* (Appendix 2) (Unpublished doctoral thesis), University of Sheffield. Retrieved from http://www.james-mooney.co.uk/publications (accessed May 15 2011).

[2] James, S. (2016). A Multi-Point 2D Interface: Audio-rate Signals for Controlling Complex Multi-Parametric Sound Synthesis. *Submitted to New Interfaces for Music Expression.*

[3] James, S. (2015). *Spectromorphology and Spatiomorphology: Wave Terrain Synthesis as a Framework for Controlling Timbre Spatialisation in the Frequency-Domain* (Ph.D Exegesis, Edith Cowan University)

[4] James, S. (2015). Spectromorphology and Spatiomorphology of Sound Shapes: audio-rate AEP and DBAP panning of spectra. *Proceedings of the 2015 International Computer Music Conference*, Denton, Texas.

[5] Schumacher, M. & Bresson, J. (2010). Compositional Control of Periphonic Sound Spatialization. *Proceedings of the 2nd International Symposium on Ambisonics and Spherical Acoustics*.

[6] Wilson, S. (2008). Spatial Swarm Granulation. *Proceedings of the 2008 International Computer Music Conference*. Belfast.

[7] Cabrera, A. & Kendall, G. (2013). Multichannel Control of Spatial Extent Through Sinusoidal Partial Modulation (SPM). *Proceedings of the Sound and Music Computing Conference 2013*, Stockholm, 532-537. Retrieved from http://smcnetwork.org/system/files/MULTICHANNEL%20CONTROL%20OF%20SPATIAL%20EXTENTTHROUGH%20SINUSOIDAL%20PARTIAL%20MODULATION(SPM).pdf (accessed January 10 2015).

[8] Kim-Boyle, D. (2006). Spectral and Granular Spatialization with Boids. *Proceedings of the 2006 International Computer Music Conference*, New Orleans, 139-142.

[9] Kim-Boyle, D. (2008). Spectral Spatialization: An Overview. *Proceedings of the 2008 International Computer Music Conference*, Belfast, 1-7.

[10] Normandeau, R. (2009). Timbre Spatialisation: The Medium is the Space. *Organised Sound,* 14(3).

[11] Topper, D., Burtner, M. & Serafin, S. (2002). Spatio-Operational Spectral (S.O.S.) Synthesis. *Proceedings of the 5th International Conference on Digital Audio Effects*, Hamburg, Germany.

[12] Kendall, G. & Martens, W. L. (1984). Simulating the Cues of Spatial Hearing in Natural Environments. *Proceedings of the 1984 International Computer Music Conference*, Paris, 111-126.

[13] Howard, D. & Angus, J. (2009). *Acoustics and Psychoacoustics: Fourth Edition*. Burlington, MA: Focal Press.

[14] Peters, N., Matthews, T., Braasch, J., & McAdams, S. (2008). Spatial sound rendering in Max/MSP with ViMiC. *Proceedings of the 2008 International Computer Music Conference*, Belfast.

[15] IRCAM (Institut De Reserche Et Coordination Acoustique), Retrieved 10th Jan 2015 from http://www.ircam.fr/1043.html?&L=1

[16] Bresson, J. (n.d.). *bresson:projects:spatialisation*. Retrieved 10th Jan 2015 from http://repmus.ircam.fr/bresson/projects/spatialisation

[17] Schumacher, M. & Bresson, J. (2010). Compositional Control of Periphonic Sound Spatialization. *Proceedings of the 2nd International Symposium on Ambisonics and Spherical Acoustics*.

[18] The Institute for Computer Music and Sound Technology. (n.d.). *ZHdK: Ambisonic Externals for MaxMSP*. Retrieved 10th Jan 2015 from https://www.zhdk.ch/index.php?id=icst_ambisonicsexternals

[19] Potard, G. & Burnett, I. (2004). Decorrelation Techniques for the Rendering of apparent Sound Source width in 3D audio displays. *The 7th International Conference on Digital Audio Effects*.

[20] Neukom, M. & Schacher, J. (2008). Ambisonics Equivalent Panning. *Proceedings of the 2008 International Computer Music Conference*, Belfast.

[21] Lossius, T., Baltazar, P. & de la Hogue, T. (2009). DBAP - Distance-Based Amplitude Panning. *Proceedings of the International Computer Music Conference*, Montreal, 17-21.

[22] Chowning, J. (1971). The Simulation of Moving Sound Sources. *Journal of the Audio Engineering Society*, 19, 2–6.

[23] Surges, G. & Smyth, T. Spectral Distortion Using Second-Order Allpass Filters. *Proceedings, 10th Sound and Music Computing Conference*, 2013. Stockholm, Sweden: SMC.

[24] Everest, F. A, & Pohlmann, K. (2014). Master Handbook of Acoustics, Sixth Edition. McGraw-Hill Education, TAB.

[25] Harris, C. (1966). The Absorption of Sound in Air versus Humidity and Temperature. *The Journal of the Acoustical Society of America*, 40.

A Permissive Graphical Patcher for SuperCollider Synths

Frédéric Dufeu
CeReNeM
University of Huddersfield
f.dufeu@hud.ac.uk

ABSTRACT

This article presents the first version of a permissive graphical patcher (referred to in the text as SCPGP) dedicated to fluid interconnection and control of SuperCollider Synths. With SCPGP, the user programs her/his SynthDefs normally as code in the SuperCollider environment, along with a minimal amount of additional information on these SynthDefs, and programs Patterns according to a simple SuperCollider-compliant syntax. From the execution of this SuperCollider session, the SCPGP interface allows for the definition of higher-level Units, composed of one or several SynthDefs. These Units can then be used in the graphical patcher itself, where the user can easily create graphs of Units, set their parameters, and, where applicable, assign them Buffers and Patterns. Permissiveness is a key principle of SCPGP: once SynthDefs have been successively tested as valid SuperCollider code, the user must be able to interconnect them with no limitation regarding connector properties (signal rate, number of channels) or the order of execution on the SuperCollider tree of Nodes. SCPGP offers a range of flexible patching operations, to foster a fully fluid and open-ended experimentation from a network of user-defined SuperCollider Synths.

1. INTRODUCTION

The variety of creative uses of SuperCollider, described by its authors as "a programming language for real time audio synthesis and algorithmic composition [1]", is assessed by its initial developer, James McCartney, in the foreword to *The SuperCollider Book*. "With SuperCollider, one can create many things: very long or infinitely long pieces, infinite variations of structure or surface detail, algorithmic mass production of synthesis voices, sonification of empirical data or mathematical formulas, to name a few. It has also been used as a vehicle for live coding and networked performances [2, p. IX]".

SuperCollider has a client-server architecture: the server application, scsynth, performs the audio synthesis and processing. Its client, sclang, is the interpreter for the SuperCollider programming language itself, and sends OSC messages to the audio server. A canonical use of SuperCollider is to write code in sclang and execute it to command the DSP operations performed by scsynth. On the one hand, SuperCollider can be used as a primarily text-based creative environment, and features such as the Just-in-Time library (JITlib) [3] offer an extended flexibility for coding-driven live performances. On the other hand, sclang has a range of Graphical User Interface (GUI) features, allowing for advanced non-text-based user interactions with both sclang and scsynth [4].

The development of the graphical patcher presented in this article is motivated by one of the possible uses of SuperCollider: an advanced text-based design of personalised DSP engines (synthesizers, samplers, processers, typically expressed as SynthDef objects in the SuperCollider language), followed with modular interconnections of these engines. Widespread equivalents to the second part of this approach in the physical world are the assemblage of modular synthesizers or the combination of guitar effect pedals. Although such interconnections can be operated solely by coding, it is here assumed that in a situation involving a number of modules contributing to a global audio graph, designating one particular module, regardless of the operation to perform on it, is easier and quicker if this module is represented as a graphical object on a two-dimensional visual workspace than as a variable name in a textual environment[1].

Beyond the ability to interconnect graphically SuperCollider-designed DSP modules, the essential advantages of implementing a patcher using the sclang/scsynth couple as a backend, as opposed to programming directly in visual languages such as Max or Pd, are twofold. First, the large library of Patterns delivered with SuperCollider [7] enables to create Event-driven Synths with great flexibility and expressivity: simple or complex Patterns can control the evolution of all the parameters of a given module, including its Buffer references and input and output Buses, providing an extended dynamism to the global DSP graph. Secondly, implementing a GUI that is extrinsic to the considered programming language favours the design of patching operations that go beyond what the visual programming languages for musical creation normally permit, thus facilitating studio experimentation and performance expressivity. The body of this article presents the global architecture of the proposed environment, before developing the features and uses of the SuperCollider code in this context, and of the two main workspaces of the GUI application itself: the Unit Maker and the Unit Patcher.

[1] More generally, the pros and cons of the textual and graphical computing paradigms are highly dependent on their contexts of use. In an article on OpenMusic published in the *Journal of Visual Languages and Computing*, Jean Bresson and Jean-Louis Giavitto affirm that "visual languages make programming and the access to computer resources more productive and useful to certain user communities, willing to design complex processes but not necessarily attracted to or skilled in traditional textual programming. They are supposed to ease programming activities (e.g. limiting syntactic errors), but also contribute to a more interactive relation between the user and the programs [5, p. 364]". Bresson and Giavitto reckon that "this idea can be argued" and point in particular to one empirical study, out of the scope of creative computing [6].

2. OVERALL ARCHITECTURE

The permissive graphical patcher for SuperCollider, referred to in this text as SCPGP for convenience and clarity[2], is being developed both in the SuperCollider language itself and as a separate GUI application built from JavaScript code in Max, embedded in a JSUI (JavaScript User Interface) object including the MGraphics library[3]. The overarching principle of SCPGP is that its user should design both elementary DSP modules and Patterns for the control of dynamic Synths as code in the SuperCollider language, and that everything else – assembling Units and playing with them – should be done from the graphical interface.

Using SCPGP first requires executing a SuperCollider document, here referred to as the SuperCollider session, that contains the user-defined SynthDefs and Patterns, as well as the backend algorithm responding to the user's actions from the graphical interface. Once the SuperCollider session has been executed, the GUI application can communicate with scsynth via sclang, with simple commands sent over a network with UDP (figure 1).

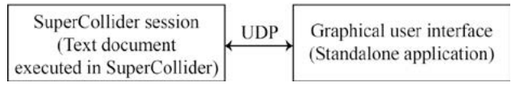


Figure 1. Overall structure of the SCPGP environment

From one given SuperCollider session (i.e., from one set of defined SynthDefs and Patterns), the user can create, save, and restore one or several GUI sessions. Creating a GUI session essentially dumps the SynthDef and Pattern information from SuperCollider to the GUI application, where the user can make her/his own Units and patch them together to generate sound and experiment.

One important aspect of the SCPGP design is that the user does not play by patching strictly SynthDef-based modules, but with Units that she/he must make from one or several of the SynthDefs declared in the SuperCollider session. The reason for this design is that some DSP modules cannot be sensibly conceived from only one SynthDef. For instance, a Unit supposed to read grains from an incoming audio stream requires the definition of two SynthDefs: the first SynthDef defines a template to record continuously the incoming audio stream into a Buffer, while the second SynthDef defines a template for reading a grain from that Buffer, with the desired parameters and envelope. A pattern can then trigger grains as dynamic and self-freeing instances of Synth that refer to the second SynthDef, while the recording Synth, referring to the first SynthDef, remains permanently active from the creation to the destruction of the unit. Therefore, the GUI application of SCPGP is itself constituted with two distinct and non-simultaneous workspaces: the Unit Maker, in which the user chooses SynthDefs and assembles them into Units, and the Unit Patcher, in which the user actually generates sound by patching together her/his previously defined Units, controls their parameters and, where appropriate, assigns them Buffers and Patterns. Figure 2 summarizes the workflow in SCPGP.

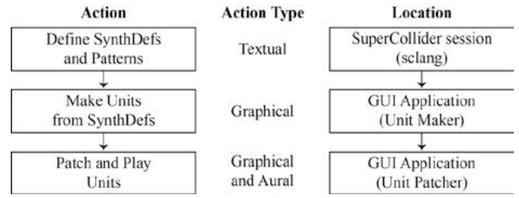


Figure 2. Workflow within SCPGP

3. THE SUPERCOLLIDER SESSION

The SuperCollider session is a text document read by the SuperCollider IDE. At the top of the document are the user definitions: essentially, SynthDefs and, if needed, Patterns. Some general settings can also be edited in that part of the document[4]. The rest of the text document should not be edited: it contains the algorithm responding to the user's actions from the GUI application. Some aspects of its implementation are described in the paragraphs dedicated to the Unit Maker and the Unit Patcher; in this paragraph are explained the declaration modes for SynthDefs and Patterns.

3.1 SynthDef declaration

In the SuperCollider session, SynthDefs are declared sequentially within a function named "func_defineUserSynthDefs" (figure 3). For each SynthDef, the user must provide a unique name (e.g. below 'Stereo Dac', 'Loop Sampler'), then declare the SynthDef as he would normally do in SuperCollider, by calling on the SynthDef class the implicit "new" method, with the name of the SynthDef and the UGen graph function as arguments. The SynthDef is then added to the SynthDescLib[5] and sent to scsynth with the "add" method.

A function named "func_initSynthDefInfo" must then be evaluated with the name of the SynthDef as an argument: this function queries the SynthDescLib to provide the SynthDef information needed by the GUI application of SCPGP[6]. As some information cannot be inferred from the SynthDescLib, the user must provide manually an

[2] The product name for SCPGP is yet to be chosen and will be given on the public release of its first beta version.
[3] A first prototype was presented by the author of this article in 2012 at the French annual computer music conference in Mons [8]. The graphical user interface was then developed with OpenGL in Max.
[4] For instance, the UDP addresses and ports for communication with the GUI application.
[5] The SynthDescLib is a library of descriptions of SynthDefs.
[6] The collected SynthDef information is: static or dynamic status (deduced from the "hasGate" member of the SynthDef description), inputs and outputs properties (audio or control rate, number of channels), argument names.

array of buffer references containing, for each reference, the control name for the buffer and its number of channels[7].

```
func_defineUserSynthDefs = {
    var all_synthdefs_info = [];
    var synthdefname;
    var synthdefinfo;

    /*************************************************************/

    synthdefname = 'Stereo Dac';

    SynthDef(synthdefname, { |in0, amp = 1|
        var signal = In.ar(in0, 2) * amp;
        Out.ar(0, signal);
    }).add;

    synthdefinfo = func_initSynthDefInfo.value(synthdefname);
    synthdefinfo.bufferControls = [];

    all_synthdefs_info = all_synthdefs_info.add(synthdefinfo);

    /*************************************************************/

    synthdefname = 'Loop Sampler';

    SynthDef(synthdefname, { |out0, bufnum, rate = 1|
        var signal = PlayBuf.ar(1, bufnum, BufRateScale.kr(bufnum) *
rate, 1, 0, 1, 0);
        Out.ar(out0, signal);
    }).add;

    synthdefinfo = func_initSynthDefInfo.value(synthdefname);
    synthdefinfo.bufferControls = [\bufnum, 1];

    all_synthdefs_info = all_synthdefs_info.add(synthdefinfo);

    /*************************************************************/
```

Figure 3. SynthDef declaration in a SuperCollider session of SCPGP for a Dac and a simple Loop Sampler.

At this stage of the SCPGP workflow, it is the responsibility of the user to ensure that the SynthDef declaration is valid SuperCollider code, and that the bufferControls array is conform to the SynthDef UGen graph.

3.2 Pattern declaration

The Pattern declaration in SCPGP is more specific than the SynthDef declaration, and takes place in a function named "func_defineUserPatterns" (figure 4). Each Pattern is initialised with a function named "func_initPatternInfo" that takes a unique name as argument (e.g. below 'Grains 1', 'Play Sample Once'). The user can then write sub-Patterns as members of the "pattern_info" dictionary[8]: first, a sequence for the durations of successive Pattern Events ('dur'); then, sequences for input and output Buses, and, where appropriate, for Buffer references; finally, sequences for Synth parameters.

At this point, the Patterns for input and output Buses and for Buffers do not take actual Bus and Buffer objects lists as arguments: rather, they take abstract indexes that will be updated when called from the GUI application. In the example of the 'Grains 1' pattern in figure 4, the Pattern design assumes that the Unit referring to this pattern can receive its input signal ('in0') from two different buses (0, 1) and send its output signal ('out0') to three different buses (0, 1, 2). When played, the Pattern will generate a succession of grains receiving signals from and sending to the actual Unit buses as follows: 1 (in: 0, out: 0), 2 (in: 1, out: 1), 3: (in: 0, out: 2), 4: (in: 1, out: 0), 5: (in: 0, out: 1), 6 (in: 1, out: 2). The same principle applies to buffer references. The Patterns for durations and parameters take lists of actual parametric values.

```
func_defineUserPatterns = {
    var all_patterns_info = [];
    var pattern_info;

    /*************************************************************/

    pattern_info = func_initPatternInfo.value('Grains 1');
    pattern_info.type = \on;
    pattern_info.dur = Pseq([0.5, 0.1, 0.2, 0.3], inf);
    pattern_info.buses.in0 = Pseq([0, 1], inf);
    pattern_info.buses.out0 = Pseq([0, 1, 2], inf);
    pattern_info.parameters.freq = Pseq([440, 550], inf);
    pattern_info.parameters.len = Prand([0.5, 0.6, 0.7], inf);
    pattern_info.parameters.amp = Prand([0.1, 0.2, 0.3, 0.4], inf);

    all_patterns_info = all_patterns_info.add(pattern_info);

    /*************************************************************/

    pattern_info = func_initPatternInfo.value('Play Sample Once');
    pattern_info.type = \on;
    pattern_info.dur = Pseq([0], 1);
    pattern_info.buses.out0 = Prand([0, 1]);
    pattern_info.buffers.bufnum0 = Prand([0, 1, 2]);
    pattern_info.parameters.rate = 1;

    all_patterns_info = all_patterns_info.add(pattern_info);

    /*************************************************************/
```

Figure 4. Pattern declaration in a SuperCollider session

When a user-defined Pattern is called from the GUI application of SCPGP, the Pattern information is used to construct and interpret a Pbind object that can then be used to play the appropriate Synth with the actual Buses, Buffers, and parameters of the designated Unit.

Here again, it is the responsibility of the user to ensure that the Pattern declaration is valid SuperCollider code. When ready with SynthDef and Pattern declarations, the user can execute the whole SuperCollider session document to interpret its code. From now on, all user actions take place in the GUI application.

4. THE UNIT MAKER

4.1 Creation of a GUI session

If no GUI session has been previously created and saved, the user must create a new GUI session from the Unit Maker. This action simply asks SuperCollider to dump to the GUI application its SynthDef and Pattern information. The left sidebar of the Unit Maker is then populated with graphical representations of template Inputs, Outputs and SynthDefs (under forms visible in figure 5a below). These templates are the constitutive elements of a new Unit[9].

Each SynthDef is represented with its name as specified in the SuperCollider session. At the top of the rectangle are its inputs: red rectangles are audio rate inputs; orange rectangles are control rate inputs. At the bottom of the rectangle are its outputs. The width of the inputs and outputs represent their number of channels. The distinction between static and dynamic SynthDefs is apparent in figure 5a. Static SynthDefs ('Buffer Recorder', 'Reverberation', 'Filter') are those that are permanent from the creation to the destruction of the parent Unit; dynamic SynthDefs ('Granulator 1') are those that are Event-driven from Patterns. Each input and output of a dynamic SynthDef can have any number of, respectively, virtual inputs and virtual outputs, so that the driving Pattern can receive from and send to different Buses, as mentioned in paragraph 3.2. Virtual connectors are displayed at the edge of a tree originating in the SynthDef connector.

4.2 Edition of a Unit graph

After creating a new Unit, the user can design its graph by clicking and dragging template Inputs, Outputs, and SynthDefs from the left sidebar of the Unit Maker to the central workspace[10]. Standard mouse and modifiers configurations facilitate a fluid patching[11]. Figure 5 shows an example of a Unit graph (figure 5a) and the representation of the corresponding Unit as to be used in the Unit Patcher (figure 5b). Inlets and Outlets are numbered automatically according to their left-to-right order; likewise, SynthDefs are labelled with their relative order of execution on the SuperCollider tree of Nodes.

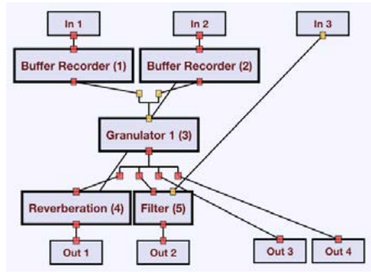


Figure 5a. A Unit graph with 3 Inputs, 4 Outputs, 4 static SynthDefs, and 1 dynamic SynthDef.

Figure 5b. The graphical representation of the resulting Unit, as to be used in the Unit Patcher

Figure 5a reveals permissiveness as one essential principle of SCPGP. Nothing prevents the user to patch any output into any input. An audio rate (red) output can send signal into a control rate (orange) input and conversely; a stereo output can be patched directly to a 4-channel input[12]; an audio output can be connected to the audio input of a SynthDef that is not below the origin SynthDef in the DSP graph (leading to the implicit creation of a block-size feedbacker)[13]; and it is possible to have as many cords from one output or to one input as needed[14]. All the corresponding signal conversions, block-size delaying for feedback, and mixing are handled by implicit Synths, automatically compiled when the entering the Unit Patcher, and are entirely transparent to the user.

4.3 Edition of the bypasser graph of a Unit

In the Unit Patcher, all Units can be bypassed by ctrl-clicking them[15]. In the Unit Maker, the user can define a specific bypasser graph for Units. When going to the bypasser graph editor, the workspace shows the Unit graph with only its Inputs and Outputs. The user can then drag cords between those to define the Unit signal graph when bypassed.

4.4 Edition of the arguments of a Unit

The right sidebar of the Unit Maker displays the parameter arguments of the edited Unit. By default, these arguments are those of the static SynthDefs constituting the Unit, excluding those relative to Buses and to Buffers. The arguments of the dynamic SynthDefs are not displayed: they are to be handled entirely by Patterns.

In some cases, the user might want to map her/his own parameter names to the SynthDef arguments. A simple example of such a case is a Unit made of four parallel 'oscillator' SynthDefs, only taking 'frequency' as a parameter argument. Rather than having four 'frequency' arguments for the unit, it may be useful to only have one 'BaseFrequency' parameter and one 'Detune' parameter.

The Unit argument editor of the Unit Maker is a text-field in which the user can declare parameters as keywords, and then type formulas to map them to the low-level arguments. Following the example described above, the user can type the lines of text as in figure 6.

```
parameter BaseFrequency
parameter Detune
frequency[0] = BaseFrequency + (0 * Detune)
frequency[1] = BaseFrequency + (1 * Detune)
frequency[2] = BaseFrequency + (2 * Detune)
frequency[3] = BaseFrequency + (3 * Detune)
```

Figure 6. Example of argument mapping

In the Unit Patcher, each Unit of this type will then appear with value fields for 'BaseFrequency' and 'Detune'. Any formula that is valid SuperCollider code can be used

[7] In the example of figure 3, the bufferControls array is empty for the 'Stereo Dac' SynthDef, that has no buffer reference in its UGen graph function, and contains one control name ('bufnum') referring to a one-channel buffer for the 'Loop Sampler' SynthDef.
[8] In figure 4, Patterns are represented with the Pseq and Prand classes.
[9] As there is typically a large number of template SynthDefs in a session, the user can also display the template items as a standard text tree and create her/his own categories of SynthDefs to navigate more conveniently.
[10] As Units are in many cases derived from one single SynthDef, a button also enables the direct creation of a Unit from one SynthDef.
[11] These configurations enable: multiple object and/or cord selection (with shift), multiple selection with a selection rectangle, copy of selected objects (with alt), copy of selected objects with copy of the cords of the copied objects (with cmd/ctrl). Connecting the inlet of an object to the outlet of another or the same object, or conversely, is done by clicking on the first connector and clicking on the second connector. By cmd/ctrl-clicking and dragging the virtual connector of a dynamic SynthDef, the user can increase or decrease its number of virtual inputs or outputs. Renaming an object with an existing Input, Output, or SynthDef name replaces it with the corresponding object and maintains the patch cords.
[12] The built-in behavior of number of channels conversion in SCPGP depends on the ratio between the number of channels of the source and the number of channels of the destination. Should the user need a specific behavior, the SuperCollider algorithm is flexible enough to be changed with a minimal of amount of recoding. It is also possible to create SynthDefs with specific channel conversion behaviors and use them explicitly in the Unit graph.
[13] In figure 5a, the reverberation is fed back into one of the recorders.
[14] The SuperCollider design of SCPGP is such that one output writes to one unique bus whatever the number of destinations, but one input reads from several distinct buses (one bus per origin). Implicit mixer Synths are created when an input needs to read from more than one bus.
[15] Including Units with only inputs or Units with only outputs, for which there is no bypasser graph, and bypassing means muting.

for the mapping. The Unit argument editor also allows to set minimum, maximum, and default values, as well as user-readable names for all parameters.

All the information of the Unit Maker (Categorisation of Units and template SynthDefs, Unit graphs, bypasser graphs, edited argument) can be saved for later restoration from the GUI application[16]. When ready with the Units from a new or restored GUI session, the user can go to the Unit Patcher of the GUI application to patch and play her/his Units.

5. THE UNIT PATCHER

The Unit Patcher is where the user actually generates sound by sending commands to SuperCollider via the GUI application. When the Unit Maker configuration has been modified (i.e., some Units have been created and/or edited) and the user goes to the Unit Patcher, the GUI application dumps information on all its Units to the SuperCollider session. The SuperCollider algorithm then makes a database of Unit types, so that any command sent from the Unit Patcher is as efficient as possible for use in a real-time critical context.

5.1 Patching Units together

As for the Unit Maker, the left sidebar of the Unit Patcher is a template palette from which the user can drag and drop items to the central workspace. However, this sidebar only contains Units[17]: while patching, the user only considers Units, and SynthDefs are transparent. Apart from the type of handled objects, patching operations are identical in the Unit Patcher and in the Unit Maker (creation, selection, move, copy, deletion) and patching is entirely permissive: the output of a Unit can be patched into the input of any Unit, including itself, regardless of signal rates, numbers of channels, relative positions on the DSP graph, number of already incoming signals. The GUI application sends compact messages over the UDP network to the SuperCollider session, which then handles Group, Synth, and Bus creations, modifications, and deletions. For patching, these messages are as follows:

- createUnits [Unit type, position on SC graph];
- deleteUnits [Unit index];
- moveUnits [Unit index, new position on SC graph];
- createConnections [Origin Unit index, Origin Output index, Destination Unit index, Destination Input index];
- deleteConnections [Origin Unit index, Origin Output index, Destination Unit index, Destination Input index].

Each of these commands can take any number of arguments, and commands can be combined into one single message to SuperCollider. Therefore, the design of SCPGP has a built-in distinction between user actions in the Unit Patcher workspace and updates of the SuperCollider DSP graph. This enables the user to choose between two main patching modes: in the direct mode, the DSP graph is updated on each user action on the Unit Patcher (as would happen in Max or Pd). In the indirect mode, the user can perform several successive actions and only update the DSP graph with all modifications happening in one go by clicking an "update" button. This enables direct transitions between significantly different DSP scenes.

While the user acts solely upon Units and patch cords, the SuperCollider interprets the commands by handling the DSP tree reordering and the instantiation of transparent Synths and Buses: figure 7 shows a diagram of all the Synths created in SuperCollider (figure 7a) given the graph as seen by the user in the GUI application (figure 7b).

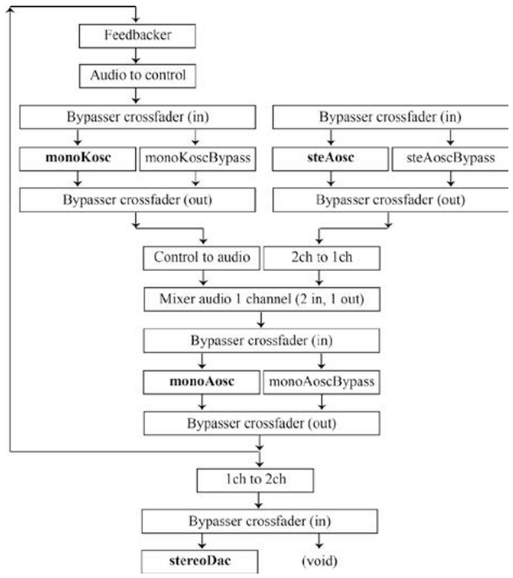


Figure 7a. A Unit graph as deployed in SuperCollider. Each box represents one Synth. In bold are the Synths corresponding to the core SynthDef of a given Unit.

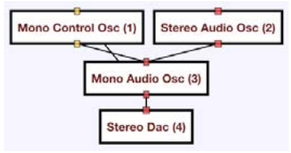


Figure 7b. The same Unit graph as defined by the user in the Unit Patcher

In addition to the direct and indirect modes of patching, some patching operations facilitate fluid changes in graph configurations. Those are especially useful when used in the direct patching mode, as they go beyond what is possible as one single action in standard visual programming environments for sound and music. These operations include:

- delete selected Units but maintain the cords going through them. In the example of figure 7b above, the user can for instance delete the "Mono Audio Osc" Unit – consecutively, the cords from "Mono Control Osc" and "Stereo Audio Osc" will be automatically repatched to "Stereo Dac", and the output of "Stereo Audio Osc" will also be patched into "Mono Control Osc".
- insert new or copied Units directly on an existing patch cord.
- rotate selected Units clockwise or counter-clockwise, maintaining patch cords. When the number of selected Units is two, this is a direct swap of both Units.

These operations are useful to change graph configurations quickly for experimentation in the studio, but are also enhancing performance expressivity: as McCartney stated in 2002, "The SuperCollider 3 Synth Server is a simple but flexible synthesis engine. While synthesis is running, new modules can be created, destroyed, and repatched, and sample buffers can be created and reallocated. Effects processes can be created and patched into a signal flow dynamically [9, p. 64]". The specific patching operations featured in SCPGP can benefit from the dynamism of scsynth: simple or complex modifications of the graph do not interrupt the signal processing, and can thus be used smoothly within a performance.

5.2 Handling Units on the graph

When a Unit has been created, the items of the right sidebar of the Unit Patcher enable a number of operations. The parameters, as defined in the Unit Maker, are modifiable via number boxes and are clipped between the user-defined minimum and maximum. As mentioned in paragraph 4.3, all Units can be bypassed by ctrl-clicking them on the Unit Patcher workspace.

Patterns are accessible as a list of names. They can be dragged and dropped onto the Pattern slot of a Unit's parameter panel: once the Pattern has been successfully checked against the internal dynamic SynthDef of the Unit, the user can simply play it and pause it with a toggle button. Here again, the GUI application is permissive. The user may have designed a Pattern with a specific SynthDef and Unit in mind, but in many cases the Pattern can be applied to another Unit that contains one or several dynamic Synths. Pattern values for existing parameter names will apply, values for non-existing parameters will simply be ignored. Parameters with no Pattern values will be played at their default values[18].

Buffers are not set in the SuperCollider session: they are allocated by the user from the Unit Patcher. There are two ways of allocating a Buffer from SCPGP: one is creating an empty Buffer by providing a number of channels and a duration in seconds, the other is to choose a sound file and fill a Buffer with it. Available Buffers can be simply dragged and dropped to a Unit's parameter panel for allocation to the appropriate Synth.

6. CONCLUSION

At the time of writing this article, the permissive graphical patcher for SuperCollider is fully functional regarding the features presented above, and is under internal alpha testing at the University of Huddersfield. The release and distribution of its first beta version will be publicly announced at the ICMC, on the presentation of this article. SCPGP offers great flexibility for those users who want both to design advanced DSP modules in SuperCollider and to interconnect them intuitively and fluidly into complex graphs. The environment can also be useful to beginners, who can focus on the UGen graph syntax of SynthDefs and adopt a modular approach immediately to test their Synths in different contexts, without having to write any code regarding Bus management.

Future work will consider user feedback following the first release of SCPGP, but two main threads are already under consideration. First, a Pattern editor will be developed in the GUI application itself: in the current state of SCPGP, Patterns cannot be modified after execution of the SuperCollider session. The Pattern editor will improve live flexibility for the control of dynamic Synths. Secondly, the implementation of a Unit Patcher scenario manager will enable the user to memorise particular patches and to navigate smoothly between her/his own previously defined DSP scenes.

7. REFERENCES

[1] SuperCollider, software homepage on GitHub, available online at http://supercollider.github.io (retrieved May 11th, 2016).

[2] J. McCartney, "Foreword", in S. Wilson, D. Cottle, N. Collins (eds), *The SuperCollider Book*, The MIT Press, 2011, pp. IX-XI.

[3] J. Rohrhuber, A. de Campo, "Just-in-Time Programming", in S. Wilson, D. Cottle, N. Collins (eds), *The SuperCollider Book*, The MIT Press, 2011, pp. 207-236.

[4] T. Magnusson, "Interface Investigations", in S. Wilson, D. Cottle, N. Collins (eds), *The SuperCollider Book*, The MIT Press, 2011, pp. 613-628.

[5] J. Bresson, J.-L. Giavitto, "A Reactive Extension of the OpenMusic Visual Programming Language", *Journal of Visual Languages and Computing*, vol. 25, no. 4, 2014, pp. 363-375.

[6] K. N. Whitley, "Visual Programming Languages and the Empirical Evidence For and Against", *Journal of Visual Languages and Computing*, vol. 8, no. 1, 1997, pp. 109-142.

[7] R. Kuivila, "Events and Patterns", in S. Wilson, D. Cottle, N. Collins (eds), *The SuperCollider Book*, The MIT Press, 2011, pp. 179-205.

[8] F. Dufeu, "Une interface graphique de manipulation d'unités modulaires dans SuperCollider", *Proceedings of the 2012 Journées d'Informatique Musicale*, Mons, 2012, pp. 123-132.

[9] J. McCartney, "Rethinking the Computer Music Language: SuperCollider", *Computer Music Journal*, vol. 26, no. 4, 2002, pp. 61-68.

[16] The GUI session is saved as a JSON document. When restoring a session, SCPGP checks the restored SynthDef information against the SynthDefs of the SuperCollider session; if no mismatch is detected, the restored GUI session is validated and the user can go directly to the Unit Patcher.

[17] Along with user-defined categories of Units for display and navigation convenience.

[18] Here, permissiveness is increased if the user gives consistent names to the arguments of all SynthDefs (e.g. all 'freq' or all 'frequency', all 'amp' or all 'amplitude').

Introducing CatOracle: Corpus-based concatenative improvisation with the Audio Oracle algorithm

Aaron Einbond
City University London
Aaron.Einbond@city.ac.uk

Diemo Schwarz
IRCAM/CNRS/UPMC
Diemo.Schwarz@ircam.fr

Riccardo Borghesi
IRCAM/CNRS/UPMC
Riccardo.Borghesi@ircam.fr

Norbert Schnell
IRCAM/CNRS/UPMC
Norbert.Schnell@ircam.fr

ABSTRACT

CATORACLE *responds to the need to join high-level control of audio timbre with the organization of musical form in time. It is inspired by two powerful existing tools:* CataRT *for corpus-based concatenative synthesis based on the* MUBU *for* MAX *library, and* PYORACLE *for computer improvisation, combining for the first time audio descriptor analysis and learning and generation of musical structures. Harnessing a user-defined list of audio features, live or prerecorded audio is analyzed to construct an "Audio Oracle" as a basis for improvisation.* CATORACLE *also extends features of classic concatenative synthesis to include live interactive audio mosaicking and score-based transcription using the* BACH *library for* MAX. *The project suggests applications not only to live performance of written and improvised electroacoustic music, but also computer-assisted composition and musical analysis.*

1. INTRODUCTION

One of the most influential paradigms in recent digital music making has been the notion of "reproduction" [1]. This includes processes of transcription, such as audio mosaicking. However, it could also be extended to reproduction of musical behavior: not only imitating sound in-the-moment, but as it unfolds in time.

A notable recent technique that lends itself to audio reproduction and transcription is corpus-based concatenative synthesis (CBCS); however, still missing is a better temporal logic for organizing synthesis based on musical structure. Individual samples are selected by targeting a list of associated features, however there is no inherent connection between the descriptors of one sample and a successive sample to be concatenated. [1]

At the same time, the *Factor Oracle* (FO) algorithm [2] has proven a successful approach to realtime pattern-recognition, most notably applied musically in OMAX [3]. Could a factor-oracle-based system be used to augment realtime CBCS, permitting a predictive logic for synthesis? Our goal is to build on the wealth of timbral detail available through CBCS along with the pattern-generating capabilities of the FO to create a flexible tool for realtime synthesis, improvisation, computer-assisted composition, and musical analysis.

[1] David Wessel, personal communication, 23 March 2012.

2. PREVIOUS WORK

The approach presented here draws on some of the most versatile existing tools for realtime interaction: CATART for CBCS and the OMAX/PYORACLE for computer-assisted improvisation.

2.1 Corpus-Based Concatenative Synthesis

CBCS systems such as CATART [4] build up a database of prerecorded or live-recorded sound by segmenting it into *units*, usually of the size of a note, grain, phoneme, or beat, and analysing them for a number of sound *descriptors*, which delineate their sonic characteristics. These descriptors are typically pitch, loudness, brilliance, noisiness, roughness, spectral shape, or meta-data, like instrument class, phoneme label, that are attributed to the units, and also include segmentation information. These sound units are then stored in a database (the *corpus*). For synthesis, units are selected from the database that are closest to given *target* values for some of the descriptors, usually in the sense of a weighted Euclidean distance. The selected units are then concatenated (overlapped) and played, possibly after some transformations. CBCS has the advantage of combining the richness and nuances of recorded sound with a direct and meaningful access to specific sound characteristics via high-level perceptual or musical descriptors.

2.2 Factor Oracle

OMAX has proven a dynamic tool for combining realtime computer-performer interaction with high-level musical representation. It first requires a "learning" phase during which audio input (for example from a live performer) is recorded, segmented, and the FO structure is calculated. The "improvisation" phase follows, in which the FO recombines the recorded segments of audio to produce new permutations of material. "Learning" and "improvisation" can overlap, so that as further audio input is added, the FO is extended as a basis for later improvisation. Multiple improvisations can be generated simultaneously, polyphonically, from the same underlying FO.

Improvisation with the FO algorithm has been described in detail elsewhere [3, 5]: the central idea is that at each segment or *state* of the improvisation, the oracle can jump along forward *transitions* to states with shared context, along *suffix* links back to states with the longest shared past, or continue to the next adjacent state. The choice among these available states is determined by user-defined probabilities and thresholds.

OMAX can take as an input symbolic MIDI pitches, or live audio signal analyzed with the YIN algorithm and *Mel Frequency Cepstral Coefficients* (MFCCs), complementing the pitch estimate with a spectral description [5]. Building on this work, we introduce a more extensive and customizable list of descriptors, especially for timbral features, to facilitate computer improvisation in contexts where pitch descriptions are inadequate: "computer noise improvisation." In the tradition of CATART, we propose that user-defined and weighted descriptor choices offer powerful creative advantages over features that describe the timbre as a whole such as MFCCs, as each of them can describe a specific aspect of the sound.

2.3 Audio Oracle

The *Audio Oracle* (AO) algorithm is an extension of FO optimized for processing of audio signals [6]. FO and AO rely on parsing the incoming signal into an *alphabet* of states; however, when instead of MIDI values, continuous ranges of descriptors are used, this becomes a non-trivial task. One of the most powerful features of AO is that it uses concepts from music information geometry to calculate an ideal distance threshold based on *information rate* (IR), a measure of "the reduction in uncertainty about a signal when past information is taken into account" [7]. Units with descriptor values within this threshold are grouped into the same state, or *letter* of the oracle alphabet.

The AO algorithm has been implemented in the freely distributed PYTHON library PYORACLE [2] [7]. In addition to providing a flexible collection of code for audio processing, PYORACLE also includes the MAX patch *pyoracle_improviser*, which allows PYTHON scripts to be called using the *py/pyext* externals. [3] The resulting improvisation tool shares many features with OMAX, but now using the AO algorithm with features calculated with the *Zsa.descriptors* library including pitch, amplitude, MFCC, spectral centroid, zero-crossing, and chroma. However these features can only be selected one-at-a time and not combined. Once the desired feature has been chosen, the AO requires an initial "training" phase: an example of the audio input is analyzed for roughly one minute, in order to calculate the IR-based distance threshold that will be used to analyze audio afterwards. Afterwards learning and improvising proceed as with OMAX.

Another innovative feature of PYORACLE, shared by the SOMAX project [8], is context sensitivity: improvisation is informed both by past events in the oracle, and simultaneously by the current audio input. For example in pitch-focused music this could encourage improvisation that blends with the immediate harmonic context of a live improvisation partner.

[2] https://pypi.python.org/pypi/PyOracle/5.5
[3] http://grrrr.org/research/software/py/

2.4 How they work together in CatOracle

The key to combining CBCS with the FO or AO algorithm is to associate *units* in CATART with states of the oracle. As mentioned above, for real-valued descriptors (as opposed to MIDI) multiple states are grouped together into letters to form an alphabet. Units may also correspond to multiple states: while this would not occur with a live input, where each new unit is unique, it could occur when the input is based upon a pre-recorded corpus, in which the same unit could be repeated multiple times (see Figure 7(b) below). These correspondences between units, states, and letters are stored in PYTHON arrays and MAX *coll* objects. Once an AO has been learned, these data can be saved for later use so subsequent improvisations can be performed without repeating training or learning phases.

As in *pyoracle_improviser*, CATORACLE incorporates a PYTHON script with the *py* MAX object. In order to support user-defined and weighted descriptors, the PYORACLE code has been adjusted to accept an incoming list of descriptors of arbitrary length and units. Each descriptor may be weighted by the user with a *multislider* object (see Figure 2 below). During the training phase, the incoming descriptors are normalized (either based on minimum and maximum values or mean ± standard deviation) and scaled by descriptor weights before the AO distance threshold is calculated. While this straightforward approach might produce statistical infelicities if descriptors are not fully independent, it is nevertheless advantageous for the subjective control it permits. As with other features of CATORACLE, the user's creative aural judgements are favored over theoretical criteria.

CATORACLE adopts the approach to context-sensitivity implemented in PYORACLE, but taking advantage of CATORACLE's extended descriptors for timbrally rich music. During improvisation, the list of next available oracle states is filtered based on a comparison with the descriptors of the incoming audio signal. Only states falling within a chosen descriptor distance, the "query threshold," are permitted for the oracle's next jump.

3. IMPLEMENTATION

After evaluating several potential architectures, it was decided that that CATORACLE would be implemented with the MUBU library for MAX and PYORACLE. This offers the efficiency and modularity of MUBU with the easy legibility and customizability of PYTHON code.

3.1 MuBu and PiPo

Multi-Buffer [9] is a multi-track container library, representing multiple synchronised data streams. A particular track might represent audio samples, a single audio descriptor or a vector of descriptors, markers or any other stream of numerical data associating each element of the stream to a precise instant in time.

The freely available binding of MUBU for MAX comes with a number of graphical visualisers/editors and externals that allow granular, concatenative, and corpus-based

synthesis. Paired with the PIPO (*Plugin Interface for Processing Objects*) framework, analysis of audio descriptors and segmentation can be performed in realtime or in batch on a whole collection of sound files.

We implemented CBCS in realtime in our CataRT system, [4] now rebased on MUBU and PIPO (Figure 1). [5]

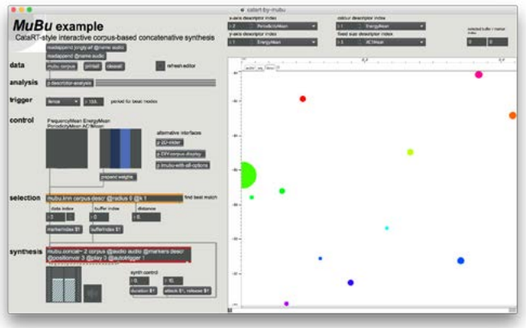


Figure 1: Screenshot of *catart-by-mubu*.

3.2 CatOracle Patch Structure

CATORACLE is distributed with MUBU in the examples folder. [6] It takes advantage of MUBU'S modular structure, with multiple objects accessing the same multi-buffer data structure through a shared argument (Figure 2).

3.2.1 Live Input

An extension of classic CBCS is realtime control using live audio to search the corpus. When descriptors are compared for closest matches between units, this could be termed realtime "audio mosaicking." Already implemented in CATART for *FTM&Co* with the module *catart.analysis* [10], this process can now take advantage of the symmetrical architecture of MUBU and PIPO for even more transparent control of identical parameters for deferred- and realtime analysis and segmentation.

In CATORACLE two segmentation methods are provided for both: "chop," which segments periodically by a specified duration; and "onseg," a simple attack detector on a specified descriptor threshold (by default based on amplitude in decibels, but reconfigurable by the user to any descriptor). The descriptors values are compared in the *mubu.knn* external, which constructs a kD-tree on the prerecorded corpus for efficient comparison with the live input to find the k-nearest-neighbors for each incoming unit. Following previous work with CATART [11], analysis can be carried out in "targeted-transposition" mode, where differences in frequency and energy between corpus and target descriptors are taken into account before re-synthesis. These data can be stored in BACH slots (see Section 3.2.5) and later edited to affect playback.

[4] http://ismm.ircam.fr/catart/
[5] http://ismm.ircam.fr/mubu, http://ismm.ircam.fr/pipo/
[6] http://forumnet.ircam.fr/product/mubu-en/

3.2.2 Audio Descriptors

By default the analysis subpatches are set to use *pipo.basic*, providing as descriptors: frequency, energy, periodicity, autocorrelation, loudness, centroid, spread, skewness, and kurtosis. This allows CATORACLE to run entirely within the free MUBU distribution. Other descriptor calculations may be customized by replacing the PIPO module with *pipo.yin*, *pipo.moments*, or *pipo.mfcc*. Or, with a software license, the full range of the IRCAMDESCRIPTORS library [12] is available with *pipo.ircamdescriptors*. [7]

A subpatcher with convenient checkboxes for descriptor selection may be substituted for the existing analysis modules in CATORACLE allowing access to spectral, temporal, or many other features in any combination (Figure 3).

3.2.3 Key Values

For large corpora, tagging individual sound files with metadata can be an invaluable tool for navigation: for example, to organize an orchestral sample library by instrument name. For this purpose CATORACLE includes the subpatch *select-by-key* to enable and disable parts of the corpus. This takes advantage of the key-value data structure of MUBU. When loading a new sound file (or folder containing sound files) to the corpus, an arbitrary key-value pair may be entered through a *textedit* object. Or the key "SoundSet" may be assigned automatically with its value set to the directory of the file, thereby allowing to group sounds beforehand. These values are saved and reloaded with the corpus. Then the sounds matching a given key-value pair can be enabled or disabled by a checkbox.

3.2.4 iMuBu View

Multi-buffers can be viewed using the graphical interface object *imubu*. Within CATORACLE, this object is accompanied by useful presets to view the waveforms of individual sound files in the corpus (wave view) with their segmentation *markers* (equivalent to units in classic CATART). Or, inspired by the CATART *lcd* view, markers may be viewed in a scatterplot with user-chosen descriptors as $x-$ and $y-$positions, $x-$ and $y-$widths, or color. Transparency is used to indicate sound files (and their markers) that have been disabled by key-value (Figure 4). From both wave view and scatterplot view, a mouse or other controller can be used to select markers for playback through *mubu.concat*~.

3.2.5 BACH Transcription

In previous work, CATART was connected to the BACH library [8] to build a DAW-like interface for concatenative synthesis based on musical score notation [13]. A similar procedure was implemented in CATORACLE: in summary, units or markers are represented as note heads on a musical staff, using either *bach.roll* or *bach.score*. Along with frequency mean (pitch), energy mean (dynamic), and duration, any other descriptor data and metadata can be saved with each note in its *slots*. In particular, the indices

[7] http://forumnet.ircam.fr/product/max-sound-box-en/
[8] http://www.bachproject.net

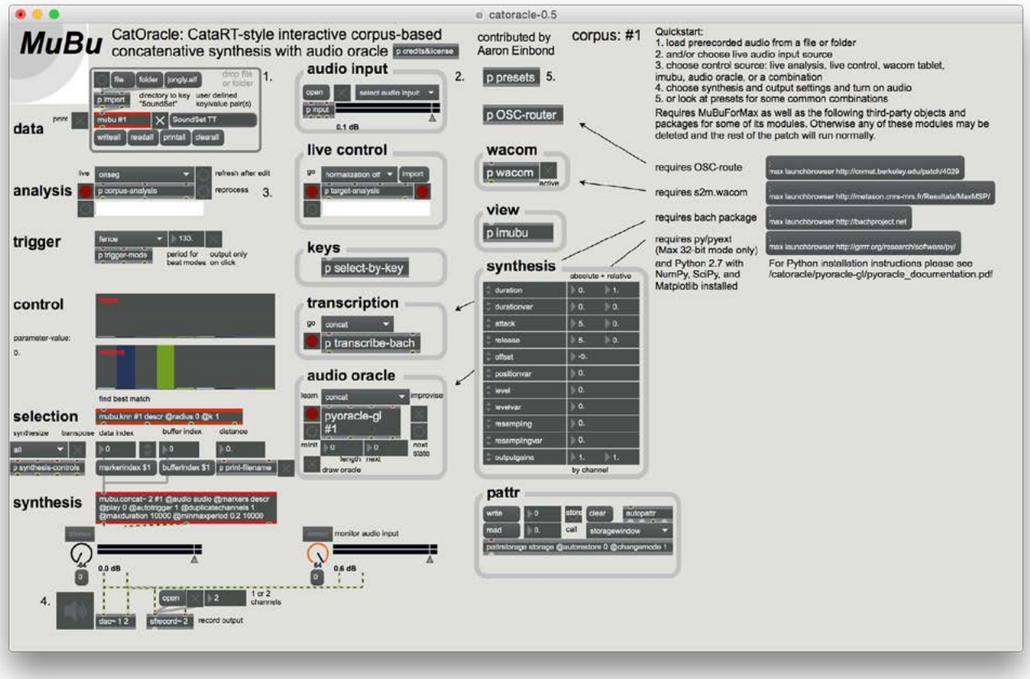


Figure 2: Screenshot of CATORACLE main patch.

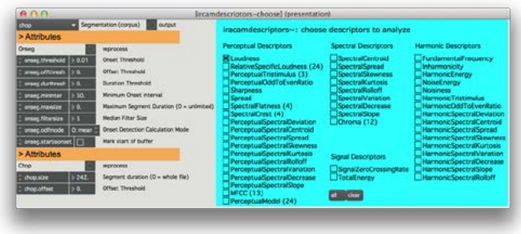


Figure 3: *pipo.ircamdescriptors* analysis subpatch.

of the marker and buffer are saved with each note, permitting playback from BACH through *mubu.concat*~. This information and other slot contents, for example source filename, can be displayed directly in the roll or score. Checkboxes permit quick selection of permitted rhythmic values with *bach.quantize*. Taking advantage of *bach.score's* proportional spacing attribute (@spacingtype 2), the roll and score are aligned rhythmically by default (Figure 5). From *bach.score* a MusicXML file can be exported, including slot metadata like dynamics and textual annotations, for further editing (see corresponding passage in Figure 9)

Combined with the audio oracle, this interface now offers new potential improvisation scenarios. For example, a computer improvisation can be transcribed in music notation for later use in computer-assisted composition (see Section 4.2 below). Or a transcription, as it is generated in real time, could be read by a human instrumentalist for acoustic playback (see Section 5 below).

3.2.6 Audio Oracle

The agent for computer-assisted improvisation is contained in the abstraction *pyoracle-gl*. As described above, descriptor values are received from other modules in the patch (*pipo*, *mubu.knn*, or *imubu*) depending on the scenario. They are normalized and weighted before being sent to the AO. The "queryae-gl" script loaded in the *py* external calls functions from the PYORACLE library to calculate the ideal distance threshold, learn the oracle, and generate the next state for improvisation. The module features a number of control parameters common to OMAX and PYORACLE: the probability of linear continuity versus jumping along the oracle, restriction to a region of the oracle, and forbidding repetition of the n most recent states with the "taboo" parameter (Figure 6). Due to the hybrid nature of CATORACLE the timing of improvisation can be controlled in several ways: durations can be reproduced from the durations of the learned oracle, durations can be taken from the pre-recorded corpus (possibly affected by *mubu.concat* synthesis attributes), or the oracle can wait to be triggered externally to advance to the next state.

The oracle can be visualized graphically using *Jitter* OpenGL objects for computational efficiency. These images, inspired by OMAX and PYORACLE show incisive views of musical structure, with forward transitions above

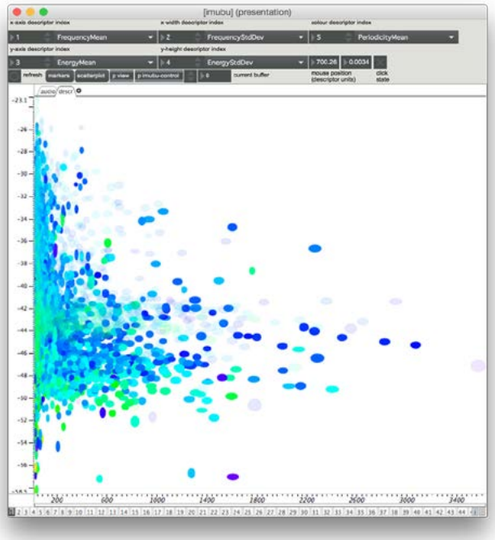


Figure 4: *iMuBu* scatterplot showing a corpus with some sound files (and their markers) disabled (transparent).

and suffix links below. The shaded ball represents the current state of an improvisation, and the shaded rectangle corresponds to a region to which improvisation is restricted (Figure 6).

3.2.7 Additional Features

Further features improve the user interface and performance of CATORACLE: communication through OSC messages using *OSC-route*, [9] control of *imubu* with a WACOM tablet using the *s2m.wacom* external, [10] *attrui* objects to control the granular-synthesis-style parameters of *mubu.concat*, and *pattr* objects with bindings to these attributes as well as other important parameters in the patch for convenient saving and reloading of preset scenes.

3.3 CataRT-MuBu-Live

An additional "light" version of the patch, entitled *catart-mubu-live*, is made available without the Audio Oracle algorithm and with no dependencies on any third-party libraries and externals. The remaining patch, requiring only MUBU and the standard MAX distribution, still retains the other features of CATORACLE, notably live analysis of an incoming audio signal for audio mosaicking, live recording the corpus, and an expanded list of triggering methods, as well as the tagging system provided by key-value pairs in MUBU. Furthermore, it avoids the limitation of the *py* external to 32-bit mode, and so can be used with MAX in 64 bits. It complements the even more streamlined *catart-by-mubu* and more elaborate CATORACLE, and all three are distributed in the MUBU examples folder.

[9] http://cnmat.berkeley.edu/downloads
[10] http://metason.cnrs-mrs.fr/Resultats/MaxMSP/

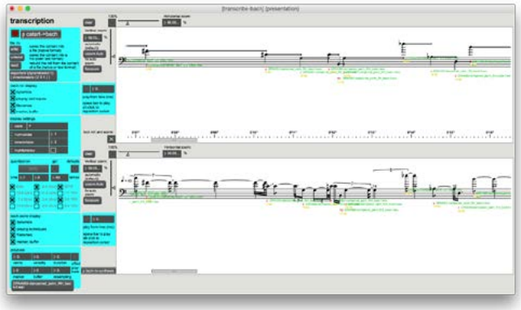


Figure 5: Transcription subpatch showing markers displayed in *bach.roll* and *bach.score* with slots for metadata.

4. MUSICAL APPLICATIONS

A range of applications extend the existing capabilities of CATART, OMAX, and PYORACLE as outlined in Figure 7.

(a) Depicts a process similar to OMAX: the performance begins with an empty corpus and the oracle is learned from a live audio input, stocking both the audio corpus and the oracle structure upon which improvisation is to be based.

(b) Represents a variation taking advantage of CBCS: a pre-recorded corpus is used in place of a live input. The oracle is learned from a musical sequence generated from the corpus, activated by a gestural controller such as a mouse or WACOM tablet. No new audio is recorded, but the oracle is recorded and used to generate an improvisation based on the same corpus.

(c) Combines (a) and (b): again the process begins with a pre-recorded corpus. But instead of a gestural controller, live audio input is used to control the initial musical sequence: for example through a live audio mosaic, comparing the live input to the closest matches in the corpus. No new audio is recorded, but the recorded oracle captures the structure of the input in terms of its descriptors. This could be advantageous in a performance situation where realtime control is desired, but without the risk of recording audio in non-ideal conditions (see *Xylography* below). Or it could be used for a more radical interpretation of computer improvisation: to imitate the behavior of one musical sequence using completely different sound material, raising intriguing æsthetic as well as technical questions.

(d) Begins with an audio oracle generated through any of the previous methods. But when improvisation begins, a live audio input is taken as a guide for navigation using PYORACLE's "query mode," so that the improvisation is informed by the current audio context. For noise-improvisation, this could be used to guide the computer improvisation toward timbral fusion with the live input, especially effective with the expanded list of timbral descriptors available from *pipo.ircamdescriptors*. [11]

4.1 Comprovisation

The combination of pre-composed music with computer-assisted improvisation, or "comprovisation" [5], is well-

[11] See a video of context-sensitive noise improvisation with CATORACLE by violist Nils Bultmann at https://vimeo.com/157177493.

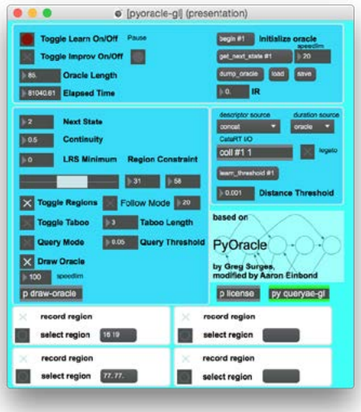


Figure 6: Audio Oracle abstraction showing (above) improvisation controls and (below) oracle visualization.

suited for CATORACLE. The first work to use the system for composition and performance is *Xylography* for violoncello and electronics by Aaron Einbond. [12] In this rigorously-composed work, no audio is recorded live: all of the electronics are generated from samples pre-recorded in the studio. However there is still a high degree of interactivity: audio oracles are learned in realtime, responding to the performer's fleeting variations in timbre and timing, especially relevant in a score with extended instrumental techniques. When the computer takes this as a basis for improvisation, it is informed by the performer's unique interpretation of the score. At times query mode is used to bring these improvisations into closer proximity with the performer as she continues playing from the notated score.

4.2 Computer-Assisted Composition

Xylography also makes use of computer-assisted composition: applying the notational capabilities of BACH,

[12] Written for Pierre Morlet and Séverine Ballon; videos available at http://medias.ircam.fr/xfb3c40 and https://vimeo.com/137971814.

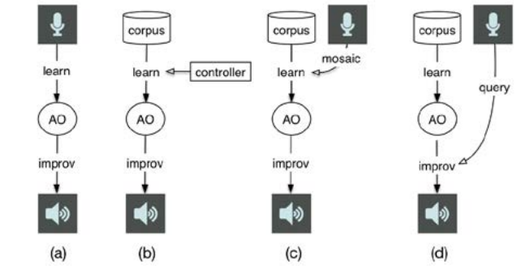


Figure 7: Paradigms of improvisation with CATORACLE.

computer-improvised sequences were transcribed as the basis for parts of the score to be performed acoustically. In this way, computer-improvisation becomes a technique for elaborating and developing acoustic material. In Figure 9, the first passage from the opening of the work is transcribed precisely from a recorded improvisation by the human performer. The second is transcribed from a computer-improvisation based on this recording, to be reinterpreted live by the performer near the end of the work. The intended effect is of a recapitulation, recognizable timbrally, but as if mis-remembered in its temporal sequence. The repetition and permutation of similar elements can be observed in spectrograms of the learned 'cello passage and the computer-improvised response (Figure 8), as well in the score, which has been edited in FINALE to render the graphical symbols (Figure 9).

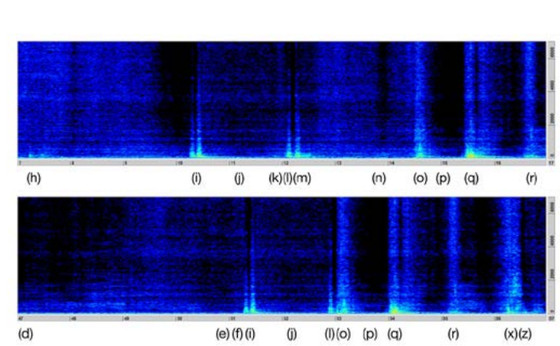


Figure 8: Spectrograms of learned and computer-improvised passages labeled with oracle states/markers.

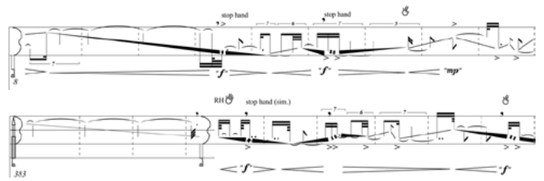


Figure 9: *Xylography* for 'cello and electronics, excerpts corresponding to those in Figures 5 and 8.

5. DISCUSSION AND FURTHER DIRECTIONS

A number of directions for further research could extend the implications of this project further.

The possibility of transcribing improvised sequences in music notation to be re-interpreted by a human performer in realtime has not yet been implemented in existing computer-assisted improvisation platforms. While the BACH package offers promising possibilities, further development will be necessary to refine the notation of dynamics, playing techniques, and realtime rhythmic quantization before it is useable in performance.

CATART'S potential for soundscape texture synthesis has already been proposed [14]. Could an AO algorithm offer a more "natural" reproduction of a soundscape, in effect imitating its behavior by permitting limitless renewal of non-repetitive textures? While no additional technical apparatus is necessary, listening tests must be employed to evaluate the effectiveness of potential results.

So far FO and AO algorithms have been used predominantly for musical creation. However their capacity for musical pattern identification and data reduction could also have uses for analysis of existing music, especially electroacoustic or timbrally rich music that still offers a challenge for existing techniques. In particular, the graphical representation of the oracle could be used to visualize large-scale formal and sonic connections. CATORACLE could be integrated with existing tools for digital analysis such as INDESCRIP or EANALYSIS [15] to provide another complementary view of musical structure.

Finally, FO and AO are only two of several oracle algorithms that could be evaluated. Another recent example is the Variable Markov Oracle (VMO) [16]. Or a related project is ImproTek [17], exploring the possibility of using pre-defined structures as templates for context-sensitive improvisation. While it could rely on a tonal structure, like a jazz progression, it could also follow an arbitrary trajectory of descriptors in time. Presently implemented in OPENMUSIC, it could exchange data with CATORACLE in the form of OSC messages. These alternative algorithms should be explored to determine how their results differ from CATORACLE and how they could be musically enriching.

Acknowledgments

We gratefully thank Séverine Ballon, Pierre Morlet, Arshia Cont, Benjamin Lévy, Gérard Assayag, Jean Bresson, Mikhail Malt, Emmanuel Jourdan, Paola Palumbo, Stephanie Leroy, Pascale Bondu, Aurèlia Ongena, Jérémie Bourgogne, Julien Aleonard, Sylvain Cadars, and Eric de Gélis. This paper is dedicated to the memory of David Wessel, mentor and inspiration for this work.

6. REFERENCES

[1] N. Donin, "Sonic Imprints: Instrumental Resynthesis in Contemporary Composition," in *Musical Listening in the Age of Technological Reproduction*, G. Borio, Ed. Farnham/Aldershot: Ashgate, 2015, pp. 323–341.

[2] C. Allauzen, M. Crochemore, and M. Raffinot, "Factor Oracle: A New Structure for Pattern Matching," in *Proceedings of SOFSEM99*. Springer-Verlag, 1999, pp. 291–306.

[3] G. Assayag, G. Bloch, M. Chemillier, B. M. Juin, A. Cont, and S. Dubnov, "Omax brothers: a dynamic topology of agents for improvization learning," in *ACM Multimedia Conference*, Santa Barbara, 2006.

[4] D. Schwarz, "Corpus-Based Concatenative Synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 92–104, 2007.

[5] B. Lévy, "Principles and Architectures for an Interactive and Agnostic Music Improvisation System," Ph.D. dissertation, Université Pierre et Marie Curie, Paris, 2013.

[6] S. Dubnov, G. Assayag, and A. Cont, "Audio Oracle: A New Algorithm for Fast Learning of Audio Structures," in *Proc. ICMC*, Copenhagen, 2007.

[7] G. Surges and S. Dubnov, "Feature Selection and Composition Using PyOracle," in *AIIDE Conference*, Boston, 2013.

[8] G. Assayag, "Keynote Talk: Creative Symbolic Interaction," in *Proc. ICMC*, Athens, 2014.

[9] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi, "MuBu & Friends – Assembling Tools for Content Based Real-Time Interactive Audio Processing in Max/MSP," in *Proc. ICMC*, Montreal, 2009.

[10] A. Einbond, D. Schwarz, and J. Bresson, "Corpus-Based Transcription as an Approach to the Compositional Control of Timbre," in *Proc. ICMC*, Montreal, 2009, pp. 223–226.

[11] A. Einbond, C. Trapani, and D. Schwarz, "Precise Pitch Control in Real Time Corpus-Based Concatenative Synthesis," in *Proc. ICMC*, Ljubljana, 2012, pp. 584–588.

[12] G. Peeters, "A large set of audio features for sound description (similarity and classification)," IRCAM, Tech. Rep., 2004, unpublished.

[13] A. Einbond, C. Trapani, A. Agostini, D. Ghisi, and D. Schwarz, "Fine-tuned Control of Concatenative Synthesis with CataRT Using the bach Library for Max," in *Proc. ICMC*, Athens, 2014, pp. 1037–1042.

[14] D. Schwarz and N. Schnell, "Descriptor-based Sound Texture Sampling," in *Sound and Music Computing*, Barcelona, 2010, pp. 510–515.

[15] P. Couprie and M. Malt, "Representation: From Acoustics to Musical Analysis," in *EMS Network Conference*, Berlin, 2014.

[16] C. Wang and S. Dubnov, "Guided Music Synthesis with Variable Markov Oracle," in *AIIDE Conference*, Raleigh, 2013, pp. 56–62.

[17] J. Nika and M. Chemillier, "ImproteK, integrating harmonic controls into improvisation in the filiation of OMax," in *Proc. ICMC*, Ljubljana, 2012, pp. 180–187.

workshop SMARTER MUSIC by Arthur Wagenaar

Almost everyone has a smartphone nowadays, but they are rarely used in a truly creative way. So far, the smartphone's *musical* potential has been sadly overlooked. Yet it's sonic potential is very high - a fantastic but under-researched area of sound. Phones and pads are used as musical controllers, sometimes as simple instruments, but this use is limited to a relatively small group of electronic music lovers. Whereas a big group of people uses their phone for gaming (mostly to kill idle time), they don't as yet play music on a similar scale.
We use our phones without much conscious thought. That I think is strange, because their influence is enormous. They make us permanently reachable, permanently trackable, and ever more we tend to outsource our memories and minds to the machines in our pockets: another very important aspect of mobile technology, that I feel does not get the attention it deserves.
The workshop **SMARTER MUSIC** combines these two elements. It seeks to explore new ways in which the smartphone can be used more thoughtfully, more creatively. Phones can be so much more than the time consuming, 'empty' machines they are now. They can be given a soul, a voice. They can be used to make music.

SMARTER MUSIC is the kick-off of a larger project: a new electro-acoustic composition I'm currently setting up called UNISONO, a 30 minute piece for orchestra and audience with smartphones (to be premiered somewhere '17-'18). Based on the novel ' The Circle' by Dave Eggers, UNISONO is about the addictive power of mobile technology, the force of group pressure to join in, and the impact this has on humanity. The project is a collaboration between myself and students of Music&Technology (HKU, The Netherlands), who will be designing musical smartphone applications, based on my compositional and theatrical ideas. These new instruments are to be used by both orchestra and audience, playing along. They should be truly *creative* instruments, offering a large scala of sound possibilities, both 'bleepy' and 'natural'. The question is: who's in control...? At first the audience is in charge, playing at will, interacting with eachother and the orchestra. But gradually the phones will take over, as hidden pre-programming will become apperent and the sound will become more narrow and forcing. With sound design the piece lets us feel what technology can do to our minds in general.
Along this composition there will be a side program where on the one hand ethical issues (see above) will be discussed more literally and more deeply, and on the other hand the technical possibilities will be demonstrated outside the temporal context of the composition. SMARTER MUSIC will be the starting point of exploring these possibilities.

The workshop will consist of two parts: first, a *presentation* by myself on my ideas regarding the sonic possibilities of smartphones in general, and the way they are going to be used in UNISONO in particular. Secondly, an *open brainstorm session*, where participants of ICMC inspire each other and are invited to join us in this promising sonic field - thus giving it the extra momentum I feel it strongly deserves. Bring your phone!

Of the smartphone's many musical possibilities, the open session will focus on these two main topics:

1) The creation of *new musical instruments*, played on a smartphone: functionality, interfacing, sound design. The most important feature of any good musical instrument, apart form sounding good, is that it should have an immediately clear, limited physical input, but an unlimited sound output. Take the piano, for instance: pressing a key gives a single note - to the left, lower, to the right, higher. Very simple indeed, yet no two pianists are the same. How can a phone be used to create something that rich, but then something radically new? What are sound and interface possibilities?

2) The use of a smartphone as a *speaker*. Any reasonably large group of people can now easily be turned into a multi-speaker sound system, because 95% of them will be carrying a phone. This I think is a Walhalla for spatial sound designers and composers. Of course these speakers aren't exactly hi-fi, but this is a case where quantity can outweigh quality. With a big number of cell phones in a crowd (or audience), sound can become alive in a fantastic and new way: three-dimensional, moving and interactive, with or without that crowd consciously participating. How to control this sound flow? Do we need a new software protocol that can be installed on our phones? What about privacy issues?

ABOUT THE AUTHOR: Arthur Wagenaar (Amsterdam 1982) is a composer, sound designer and pianist. His works are theatrical by nature: the music is never 'just' an abstract compositional concept, but wants to tell something about the real world, trying to inspire the audience, and setting their minds to work through their ears. The impact of technology and the position of nature in our lives is a topic in many of his works, which are explored using both acoustic and electronic means. Recent works include *Stads/Einder (City's/Horizon),* for prepared piano and 12 overhead projectors, and *Guess Who's Back*, a theater performance by his band Susies Haarlok.
www.arthurwagenaar.nl / www.susieshaarlok.nl

photo: Baldwin Henderson

Granular Wall: Approaches to sonifying fluid motion

Jonathon Kirk
North Central College
jkirk@noctrl.edu

Lee Weisert
University of North Carolina at Chapel Hill
weisert@unc.edu

ABSTRACT

This paper describes the materials and techniques for creating a sound installation that relies on fluid motion as a source of musical control. A 4' by 4' acrylic tank is filled with water and several thousand neutrally-buoyant, fluorescent, polyethylene microspheres, which hover in stillness or create formations consistent with a variety of turbulent flows. Fluid motion is driven by four mounted propulsion jets, synchronized to create a variety of flow patterns. The resultant motion is sonified using three essential mapping techniques: feature matching, optical flow estimation, and the direct mapping from motion data to spectral audio data. The primary aim of the artists is to create direct engagement with the visual qualities of kinetic energy of a fluid and the unique musical possibilities generated through this fluid motion.

1. INTRODUCTION

The study and application of fluid mechanics covers a universal and wide-ranging array of phenomena that occur in nature and in day-to-day human activities. Natural fluid behaviors of the smallest scale to extreme magnitudes can include everything from microscopic swimming animals and blood flow to continental drift and meteorological phenomena. [1] As generative processes for sound synthesis and sonification become more accessible and creative, we feel that there exist many possibilities for sonic exploration for a whole range of fluid behaviors. *Granular Wall* is an attempt to engage directly with certain physical aspects of fluid flow–namely, specific types of turbulent fluid flow in water. Our primary point of departure is to discover compositional structures between fluid statics and fluid dynamics.

Recent research in areas related to sound-synthesis based on the physics of liquids in motion primarily has dealt with innovations in auditory display and simulations related to smoothed particle hydrodynamics. [2] While these methods continue to contribute greatly to the fields of computer graphics and animation, *Granular Wall* is an attempt to present a variety of fluid phenomena in an aesthetically-oriented and immersive sonic and sculptural form.

2. COMPOSITIONAL MOTIVATIONS

Besides a general interest in sonifying complex geometries, vortices, and chaotic processes, there were several compositional motivations for designing and building the sound installation. Iannis Xenakis's ideas related to algorithmic music served as an important analytical foundation while designing and composing with such unpredictable parameters as turbulent fluid flow. Perhaps the most interesting observation of his is that certain mechanizable aspects of artistic creation may be simulated by certain physical mechanisms or machines. [3] And certainly from a compositional point of view this process works in reverse. Many of the ideas outlined in Xenakis's *Formalized Music* eloquently describe how composers can turn to complex natural processes for the creation of musical structures. For example, sonic events can be made out of thousands of isolated sounds, and that multitude of sounds, understood as a totality, becomes a new sonic event. This mass event is articulated and forms a "plastic mold" of time, which reflects aleatory and stochastic laws. [4]

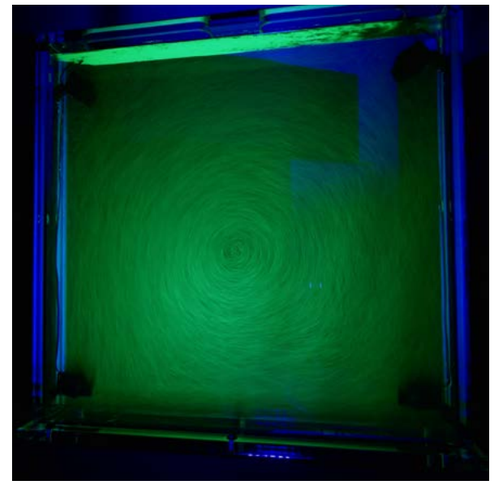


Figure 1. A time-lapse image of *Granular Wall* showing a spiral vortex created by four jets within the tank.

Another point of historical influence is James Tenney's influential *META+Hodos*, a work that puts its primary focus on how music is codified within multidimensional space. Tenney's application of Max Wertheimer's *Laws of*

Organization in Perceptual Form to music theory and analysis was a brilliantly provocative alternative to traditional score-based analytical methods. In particular, numerous musical analogies to "points and lines in the visual field," as well as the concept of using factors of proximity, similarity, and intensity for organizing musical elements allows for a seamless conceptualization between the visual and acoustic dimensions. [5] Indeed, the mechanical differences between the auditory and visual apparatus–and thusly, our comprehension of visual and acoustic material–are so fundamentally contrasting that an analytical application of evolutionary gestalt psychology is probably necessary for any "fluid" discussion of visual sonification.

In *Granular Wall*, the motion tracking of moving particle clouds, spiraling formations and traveling internal waves allows for the creation of parameters that can effectively be mapped to sonic fields: pitch, register, overall density and intensity, spatial location, morphology, timbre and the general progress of the form. For example, a spiral vortex can correspond to a range of musical ideas by sonifying its shape (Figure 1). Because of the immediate and arresting visual qualities that fluid motion provides, perhaps it is fitting that Xenakis directly speaks to fluid motion serving a musical purpose: "the archetypal example is fluid turbulence, which develops, for example, when water flows rapidly around an obstruction…[resulting in] a set of mathematical mechanisms common to many systems that give rise to complicated behavior." [6].

Furthermore, many fluid physicists are motivated not only by their important scientific goals of their work, but also by a visceral fascination with their work. [7] Van Dyke's seminal *An Album of Fluid Motion* and the annual Gallery of Fluid Motion presents fascinating and dazzling visualizations of innovative flow techniques related to both liquids and gases. [8] The striking variety and complex beauty of the visual phenomena, coupled with the sonic component offers a multi-faceted reading of the limits and capabilities of our various perceptual apparati, as well as how they can be represented in an artistic context. We also quickly realized during the development phase of the sound installation that we could re-create many sophisticated fluid flows using relatively simple visualization techniques.

3. DESIGN AND MATERIALS

In order to visualize fluid motion patterns at a scale appropriate for observers within a gallery space, we designed a custom standing tank of clear polished 3.175 cm acrylic (1.2 meter length x 20cm width x 1.2 meter height). The top is partially closed with a flanged bottom for bolting to a metal stand. The 208 liter tank then holds some tens of thousands of bright fluorescent green (505nm peak) and neutrally buoyant polyethylene microspheres (500µm) designed with density ~1g/cc for suspension in fresh water. Precision density calibration ensures that during extended periods of inactivity in the propulsion jets, the spheres saturate the tank with even dispersion, rather than gradually sinking or floating to the top. Because the spheres are manufactured to be hydrophobic, we coated them with a soap surfactant prior to suspending them in the tank. A Chauvet ultraviolet lighting system is placed in front (diagonally) of the tank so that the microspheres are maximally illuminated.

Four propulsion jets are attached with neodymium magnet suction mounts to the four corners of the tank. The location and directionality of the jets is carefully calibrated to ensure vertical orientation of fluid motion, allowing for the possibility of vertically and horizontally symmetrical flow shapes (ascending convection pattern, descending convection pattern, "four-leaf clover," right/left-facing double spirals, turbulent collisions, etc.).

The jets are driven by an Arduino-controlled electrical relay switch, which follows a pre-composed 20-minute cycle of synchronized on/off steps. Every combination of jets is represented in the relay sequence, and the ordering of the sequence is designed to highlight and contrast the various possible motion types. For example, a full-tank clockwise spiral vortex is achieved by initiating the top left and bottom right jets, and this is followed by a spiral vortex in the opposite direction (bottom left and top right jets, counterclockwise flow direction), resulting in a period of chaotic disruption before a relatively laminar flow pattern is reestablished. In another scenario all four jets are initiated for a relatively brief period of time–long enough to disrupt all of the fluid in the tank but not long enough to create a stable clover-shaped flow pattern–followed by an extended period of inactivity in which the complex interactions of the initial burst are slowly played out. By and large, the compositional work of the sound installation is located in this sequence of relay switches. The timings, durations, spatial locations, and juxtapositions of the jet motions are analogous to compositional ideas of contrast, sectional divisions, and large-scale form.

Two cameras for motion tracking are located on the opposite side of the tank from the viewers. Two laptop computers (processing the flow visualizations in real-time), audio interfaces and a mixer are all hidden underneath the tank. The resultant sound synthesis is sent to left and right channel monitors placed approximately 1 meter from each side of the tank.

4. SONIFICATION AND MOTION TRACKING

Several prevalent computer vision techniques are used to translate the various flow visualizations to sound. It was our desire to find mapping strategies that both give a direct correspondence to the directionality and velocity of flow patterns as well as a less direct sonification of the unpredictable patterns created through turbulent flow. The motion tracking is implemented using various analysis processes within the Max/MSP/Jitter programming environment. [9] Computer vision and motion tracking methods are implemented using the cv.jit library. [10] In *Granular Wall*, we were primarily interested in mapping the movement of the microspheres in a fluid medium in more than one way at once. This sonification technique is consistent with Yeo and Berger's application of image sonification methods to music, where both scanning and probing the image or video input are both used. [11] More specifically, we settled on methods that both sonified the video images in a fixed, non-modifiable order and more arbitrarily at different regions of the video image (regions of the tank).

4.1. Feature Matching and Optical Flow

Within computer vision literature, the term feature matching can refer to the analysis of visual structures defined by interest points, curve vertices, image edges, lines and curves, or clearly outlined shapes. [12] By tracking the motion of the particles either as a singular cloud shape or in several localized cloud shapes we are able to analyze the morphing images by matching different features in one frame to the most similar features in the second. [13]. One such motion flow analysis method used is the mapping of this cloud motion of microspheres within a matrix of 36 (6 x 6) subsections of the tank (Figure 2). As propulsion jets are turned on and off, flow patterns can be tracked by narrowing in on these specific subsections of the tank. We found that the Horn Schunk optical flow algorithm was the most efficient way of estimating the directionality and velocity of particles within the individual chambers. The resultant synthesis presents granular sounds moving up and down in frequency and spatialized either to the left or right channel.

One aspect of motion tracking that is distinctive to fluid motion–and in particular to the motion of neutrally-buoyant particles within a fluid medium–is the difficulty of defining region boundaries within a highly dispersed field. Neutrally-buoyant particles reflect the highly entropic nature of internal fluid dynamics, as opposed to particles of a different density than the fluid medium, which will form into more clearly differentiated shapes and patterns. While the gravity-defying qualities of the 1g/cc particles lend a striking elegance and beauty to the installation–in addition to more accurately reflecting the internal motion in the tank–it presents a challenge in terms of motion analysis. In order to successfully translate the fluid motion into digital information, the captured video image was first reduced using an adaptive threshold limiter to filter out less bright particles (the adaptive capability of the limiter was also essential due to the difficulty in achieving a perfectly even diffusion of ultraviolet light throughout the tank). After filtering out darker pixels, remaining pixels were "dilated" (new pixels were added surrounding the filtered pixels in order to make them appear larger). Finally, a visual delay effect (or "slide") was added to more prominently express the motion of the pixels over time. To do this, illuminated pixels decrease gradually in luminosity in subsequent matrix frames. The resulting image more closely resembles a group of isolated entities with clearly visible vector paths as opposed to the unprocessed image, which is closer in appearance to undifferentiated static. A careful balance needed to be negotiated during the visual processing phase so as not to reduce the image to the point that the complexity of the fluid motion was no longer communicated.

In addition to the left/right and up/down directional analysis, the relative speed of fluid motion was able to be described by measuring varying levels of overall (averaged) luminosity within each of the 36 subsections. When the fluid motion is slow, the fluorescent microspheres are more stationary and thus reflect more continuous light into the camera, and vice versa, fast-moving spheres do not reflect as much light. Thus, the overall luminosity is inversely proportional to the velocity of the fluid motion. A range of luminosity readings is typically created for each performance of the installation (depending on the ambient light in the space as well as the number of spheres and placement of the lights) and this is scaled to a range of rhythmic values for the sound synthesis. Note durations are also mapped to fluid speed, thus there is a continuous transformation between rapid, short blips and slow, gradually decaying tones.

The synthesis consists entirely of sine tones with a bell-like amplitude envelope (20 ms attack, variable-duration exponential decay). Musical parameters of pitch, panning, glissando direction, note density, and note duration are determined both by the location of the subsection of the tank and by the motion of the microspheres. The six vertical rows of the tank are divided into six octaves with low frequencies mapped to the bottom of the tank and high frequencies mapped to the top of the tank. Within each of the one-octave ranges, a pitch set built on a justly tuned scale is randomly applied to the granular synthesis tones. The frequencies for the lowest octave are: 110 Hz, 123.75 Hz, 137.5 Hz, 151.25 Hz, 165 Hz, 178.75 Hz,

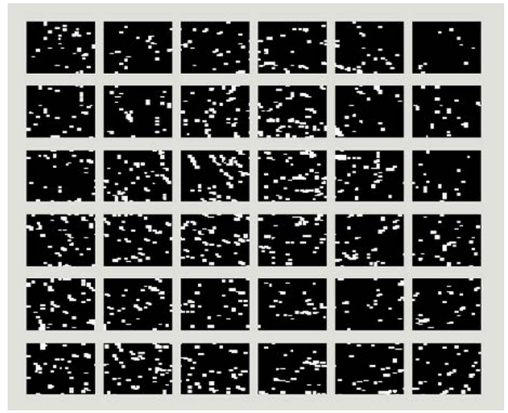


Figure 2. The separation of the tank into 36 chambers for optical flow analysis and feature matching.

192.5 Hz, 206.25 Hz. These frequencies are doubled to provide ascending octaves in each of the horizontal rows of the tank. However, due to the glissandi applied to each tone, the aggregate harmonic quality of the pitch set is only faintly discernible.

The six vertical columns are mapped to the global panning ranges for the tones as follows (from left to right): Column 1 - 100% pan left, Column 2 - 80% pan right, Column 3 - 60% pan left, Column 4 - 60% pan right, Column 5 - 80% pan right, Column 6 - 100% pan right. The glissando for each note direction is determined by the up/down direction of the microspheres provided by the motion tracking analysis. The overall speed of directional movement of the microspheres is mapped to a per-

cent deviation from the starting frequency with faster movement up or down resulting in a greater percent deviation from the starting pitch. The range of frequency deviation in the glissandi is 0% to 20% of the starting pitch. Musical parameters of note density and note duration are inversely proportional and are mapped to the sum of the pixel values of a particular subsection (range of 10,000 to 800,000). Each subsection is monophonic (allowing for 36-voice polyphony overall), with a note duration range of 50 ms to 3000 ms. Depending on the overall lighting in the gallery space, the visual threshold in the image processing must be adjusted to ensure an even displacement of note durations.

4.2. Mapping to Spectral Audio

Another method we found successful for sonifying the fluid motion types in the tank was to translate the tank directly as a visual spectrogram, where the lowest sound frequencies of one aspect of the audio output were rendered by the particle motion in the lower regions of the tank. This video analysis simply calculates the absolute frame difference between subsequent video frames. [14] The particle movement is mapped more easily using the same thresholding technique described earlier. Using this method, we are able to generate additive synthesis while considering the entire area of the microsphere movement. The inverse spectrogram approach of taking the square tank and transferring the video tracked images on the Y-axis (frequency) and X-axis (time) is also adapted partially from the sonification methods in motiongrams–where video analysis tracks a moving display as a series of motion images. [15] This is similar to the now common technique of 'drawing' a spectrogram onto a two-dimensional plane. In our Jitter implementation, the matrix data is mapped to audio via jit.peek~, and then sent to an interpolated oscillator bank (640 oscillators) to generate the drone layer. The resultant timbres are then applied to the audio mix to create a timbrally-rich musical background, which can represent the tank at its most static (stillness) and at its most dynamic (turbulent flow). Figure 3 presents a screenshot from our motion tracking sequence of the inverse spectrogram method.

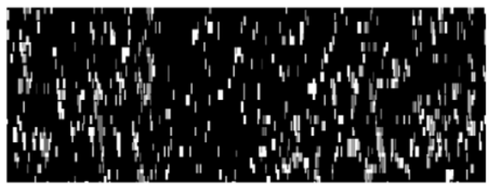


Figure 3. A video screenshot of our implementation of the inverse spectrogram technique: particle movement is translated to frequencies and amplitudes on the 2D spectrographic plane.

As we can see from the previous description of the sonification mappings, a significant reduction in information occurs during the translation from the visual to sonic realms. This is due both to the computational limitations of the real-time synthesis as well as the different perceptual capacities of the eyes and ears. With these mappings and parameter ranges, the general motions of the microspheres can be clearly discerned aurally, which is the primary goal of this aspect of the sonification process.

5. FUTURE WORK

Sonifying fluid motion has many rich possibilities both within electroacoustic composition and, more importantly, in intermedia art forms. Because there are many creative ways that artists can make fluid dynamics visible, it follows that sound artists and composers will be able to find ways to make flow visualization audible. Taking our initial motivations further, we are interested in turning the process back on itself–where sound can be used to alter the fluid flow of a liquid or gas. The use of ultrasound, surface acoustic waves, and even acoustic microfluidics offer some possible points of departure.

As composers we are interested in finding other ways of generating timbre and spatialization that corresponds to the movement of the fluid. Because of the challenging nature of mapping motion to music, and that many times we are led to make arbitrary decisions about timbre, it is important to consider a wide range of synthesis techniques that represent our chosen materials–in this case, water and fluorescent particles. Future sound processing strategies could include using audio sampled from our visual and/or physical source, and then assigning vector positions to sound file positions. [16]

6. CONCLUSIONS

Sonification of fluid dynamics presents technological challenges in the analysis and distillation of highly entropic and dispersed environments. Image reduction and compartmentalized directional flow analysis must be balanced with a computationally-intensive representation of the complex interplay of internal motions of the fluid. Sonification of such an environment is achievable through flow analysis and simulated granular resynthesis, as well as more directly via direct visual mapping to spectral audio.

A conceptual challenge–one that likely applies to most attempts at sonification of visual phenomena–is that complexity of fluid dynamics is more fully comprehensible to the eye than to the ear. Sound artists must necessarily make creative and sometimes arbitrary decisions regarding the materials and morphology of the music in order to create connections between the audio and visual realms. The most direct translation of data may not necessarily provide the clearest expression of the visual elements.

7. REFERENCES

1. Munson, B., Young, D., & Okiishi, T. (1998). *Fundamentals of fluid mechanics*. New York, Wiley.
2. Drioli, C., & Rocchesso, D. (2011). Acoustic rendering of particle-based simulation of liquids in motion. *Journal on Multimodal User Interfaces*, 5(3-4), 187-195.
3. Xenakis, I. (1971). *Formalized music: thought and mathematics in composition*. Bloomington: Indiana University Press.
4. Ibid.
5. Tenney, J. (1964). *Meta (+) Hodos*. New Orleans: Inter-American Institute for Musical Research, Tulane University.
6. Solomos, M. (2006). Cellular automata in Xenakis's music. Theory and Practice. In *International Symposium Iannis Xenakis* (pp. 11-12). Athens, Greece.
7. Hertzberg, J., & Sweetman, A. (2005). Images of fluid flow: Art and physics by students. *Journal of Visualization,* 8(2), 145-152.
8. Van Dyke, M. (1982). *An album of fluid motion*. Stanford, Calif.: Parabolic Press.
9. Puckette, M., & Zicarelli, D. (1990). *Max/MSP*. Cycling 74.
10. Pelletier, J. (2013). cv. jit–. *Computer Vision for Jitter.* http://jmpelletier. com/cvjit/. February, 15.
11. Yeo, W. S., & Berger, J. (2006). Application of raster scanning method to image sonification, sound visualization, sound analysis and synthesis. In *Proceedings of the International Conference on Digital Audio Effects* (pp. 309-314). Montreal, Canada.
12. Pelletier, J. M. (2008). Sonified Motion Flow Fields as a Means of Musical Expression. *Proceedings of the International Conference on New Interfaces for Musical Expression* (pp. 158-163). Genova, Italy.
13. Ibid.
14. Jensenius, A. R. (2006). Using motiongrams in the study of musical gestures. *ACHI 2012: The Fifth International Conference on Advances in Computer-Human Interactions* (pp. 170-175). Valencia, Spain.
15. Ibid.
16. Pelletier, Ibid.

The Computer Realization of John Cage's *Williams Mix*

Tom Erbe
UC San Diego
tre@ucsd.edu

ABSTRACT

This paper describes the process of creating a new performance of John Cage's early tape music piece Williams Mix (1952). It details the features of the score, the sound library, and the process used by Cage and his group of friends (David Tudor, Earle Brown, Louis and Bebe Barron, etc.) to construct the piece. The construction of a new version of the piece is then described, discussing the problems interpreting the score, the collection of sounds according to Cage's specification, and the creation of a computer music patch to perform Williams Mix.

1. INTRODUCTION

In the summer of 2012, I decided to attempt a new version of John Cage's second piece for magnetic tape, *Williams Mix*. My intent was to create performance software that would vary the unfixed elements and make each playing of the piece unique. As no-one had performed *Williams Mix* directly from the score since the original realization[1], I was interested in what would come from a strict adherence to the original score and notes – to find whether a new performance would resemble Cage's original rendition.

The main impediment in creating a performance of *Williams Mix* is the length and detail in the score [1]. Each of the 192 pages depicts 20 inches of 8 tracks of magnetic tape. The entire piece contains over 3000 tape splice shapes, and each shape requires at least 8 measurements. As I have recently worked on several large and complex recording projects (Lucier's *Slices* and Berio's *Duetti per due violini*) I felt ready to take on the task.

My approach was to manually record all of the information from the graphic score onto a spreadsheet. I collected a new sound library of 500-600 sounds of "all audible phenomena"[1] by making new field and other recordings, and by asking around 20 of my friends and colleagues to contribute. Finally I created a computer program in the music language Pure Data that reads the event data and selects, sequences, edits, fades, spatializes and processes the sounds according to the score markings.

[1] It should be noted that both Larry Austin [8], and Werner Dafeldecker & Valerio Tricoli [9] composed new interpretations of *Williams Mix* adopting Cage's compositional methods (although not following the score).

2. APPROACHING THE SCORE

The first task in performing *Williams Mix* was to convert the score into a form that could be read by computer software. This required identifying, measuring and noting all of the sonically relevant features on the score. I worked initially with my colleagues in computer graphics to see if the measurement process could be expedited, but it soon became apparent that the noise and complexity of the score made an automated process more of a computer research project than a solution. After scanning the score at high resolution, and adjusting the scans to match the original size of the score, I then proceeded to measure *Williams Mix* manually.

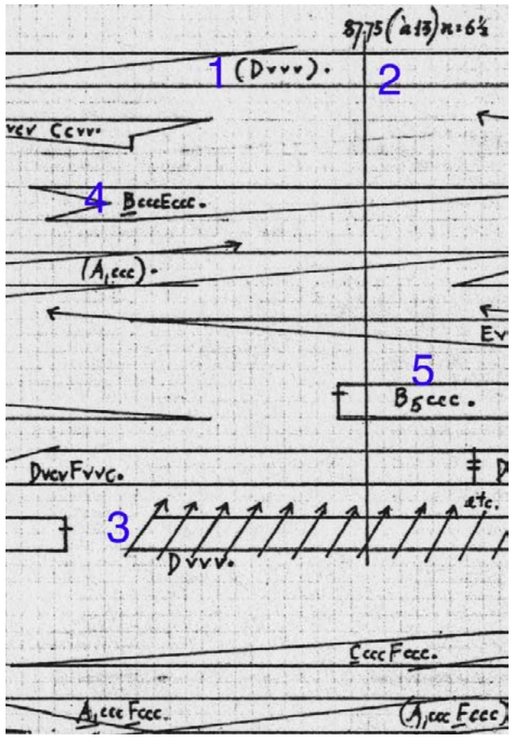


Figure 1. Features in the score: 1) Parentheses around category, 2) Section markings, 3) Crosscut splicing, 4) Looped double sound with double fade in, 5) Hash mark indicating indeterminate fade type, number indicating specific Bccc sound .

The score for Williams Mix is a diagram of the pattern of edits of 8 channels of magnetic tape. The tape is shown at 100% scale, so one could lay the tape on the score to cut the tape "like a dress-makers pattern" [2]. Each page represents one and one third of a second of sound (twenty inches per page divided by a tape speed of fifteen inches per second) and the score is 192 pages for a total length of 255 seconds (the piece ends in the first half of the last page).

Before I started work, it was important to determine what features to measure and record. Each tape splice segment has a shape, a channel (from 1 to 8), a start and end position, one or two sound categories, and several marks indicating particular editing techniques which result in various sonic transformations. This includes: a horizontal dash at the beginning and/or end of the splice; crosshatched arrows throughout or at the beginning or end of a splice, and an underline under either of the sound categories.

Cage describes most of these marks in the note to the score. There were two other marks with no explanation. First, the category is sometimes noted in parenthesis. This seems to indicate a continuation of the category previously noted following an editing technique change. Second, the category sometimes appears with an additional number. As the original tapes developed for Williams Mix also contained these numbers, I believe that this indicates that a specific sound of a given category is to be used.

Finally, there are sectional notations in the score, such as that on the top of page 5: "87.75 (á13) n=6 1/2". The first number indicates the number of inches from the start, the second - "á" - indicates the density of the section from 1 to 16, and the third - "n" - is the time base of the section (in this case, 6 1/2 inches). These numbers determine the length and densities of the splices created for each section, but are not needed to perform the score.

2.1 Splice shape feature details

Several of the splice shape features - track number, page number, start position and end position - are noted plainly. Track number ranges from 1 to 8, page from 1 to 192, start and end from 0 to 20 inches (from the start of the page). One point of interest is that measurement accuracy does affect the timing accuracy. As the line width in the score varies from 0.01 inch to 0.03 inch, a 0.03 inch accuracy is probably the best one can hope for. At the tape speed of 15 inches per second this corresponds to a 2-millisecond accuracy.

The shape of each splice is more complex. At the start of my project, I was recording the full dimensions of the polygon that made up each splice shape. I soon found that this was too time-consuming (I was not making quick enough progress to complete the piece by the debut concert date). From a quick inspection of the score, I found that almost all of the splice shapes could be reduced to a pair of trapezoids by dividing the shape with a horizontal line (parallel to the direction of the tape). With this simplification, only 8 numbers need to be recorded per shape; the horizontal position of each point in the two trapezoids. The vertical measurements are always 0 and 1/8 inch. Shapes that do not fit the two-trapezoid simplification were treated separately.

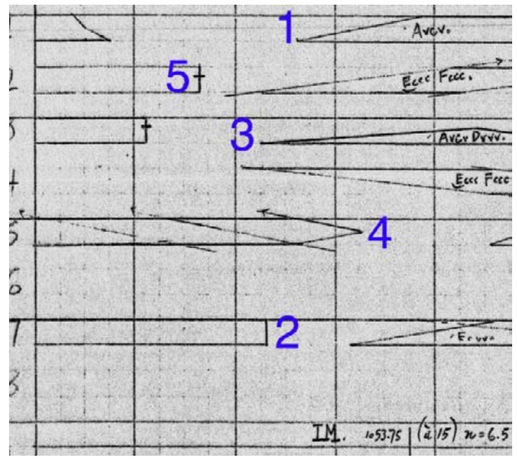


Figure 2. Fade types: 1) Simple fade in, 2) Blunt cut, 3) Fade that does not reach full volume, 4) Double fade. Note the section marking on the bottom that indicates the piece is now immobile "IM." *Score images Copyright © 1960 by Hanmar Press, Inc. Used by permission of C.F. Peters Corporation. All Rights Reserved.*

Diagonal lines indicate crosscut tape across the tape splice shape with arrows indicating the original tape direction. In the original realization of Williams Mix this required cutting the tape into rhomboids with a 1/4 inch distance between the slanted sides so that the rhomboids can be rotated and connected together. This splicing technique can cause sound fragmentation, filtering and reversal. I simply noted the angle of the arrows on the score and whether the crosscut is repeated, or only occurs once at the beginning or end of the splice shape.

Sound type is indicated with a set of four letters, and each sound type may also have a number and/or an underline. Each tape splice shape may have one or two sound types. If there are two, then two sound types are to be mixed together. The underline indicates sounds that are looped or repeated. There is no indication how quickly the looping is to be performed. As stated earlier, the number may indicate that a specific recording in the given sound type should be used. In the original sound library created by the Louis and Bebe Barron, the source tapes were similarly numbered [3].

The first letter of the sound type designates the sound category: A - city sounds; B - country sounds; C - electronic sounds; D - man-made sounds (including the literature of music); E - wind-made sounds (including song); and F - small sounds requiring amplification to be heard

with the others. The subsequent three letters are c or v, and indicate whether the pitch, overtone structure and amplitude are constant or variable. The six categories and two states for pitch, timbre and amplitude result in a set of 48 sound types.

There is no indication in the score for the amplitude of any given sound. The description for category F (small sounds requiring amplification to be heard with others) implies that the recorded material should be at a similar amplitude, and that the number of tracks that are active and whether one or two sounds are mixed in a track will determine the overall amplitude (this corresponds to the "á" number for the section).

3. THE SOUND LIBRARY

When recording and collecting sounds for *Williams Mix* one needs to determine the category and variable or constant aspects of each sound. Examination of the categories soon reveals that many sounds could fit in multiple categories. A wind-produced sound could be a manually produced sound, a country sound and a small sound at the same time. In my version, I decided to give preference to more specific categories – separating small sounds first, wind and electronically produced sounds second, manually produced sounds third, and finally city and country sounds. Similarly there will be different interpretations of "constant" or "variable" for pitch, timbre and amplitude. I decided to use similar guidelines to those David Tudor used for interpreting "simple" and "complex" in *Variations II*. [4] "Constant" describes either a fixed value or simple repetition in frequency, timbre or amplitude; "variable" describes change and unpredictability in those parameters.

Cage does not say much about the sound library in the score notes besides describing the categories, and stating "The library of sounds used to produce the *Williams Mix* numbers around 500 to 600 sounds". From an analysis of the score one can see that the sounds are fairly evenly distributed among the A, B, C, D, E and F categories and that ccc and vvv occur more often than other combinations. However, the score does not say how much repetition is allowed for a given category. If Accc occurs more than once, it could be a foghorn, then a helicopter, then a drill; or it could be a foghorn each time. In Cage's composition notes on *Williams Mix* [5], one can find more detail on the methods used to populate the sound library. Cage initially used a deck of 1024 cards to select the sounds. In this deck, for single (unmixed) sounds, there are 4 sounds with 3 variable aspects (vvv) repeated 32 times, 8 sounds with 2 variable aspects (cvv, vcv and vvc) repeated 16 times, 16 sounds with 1 variable aspect (ccv, cvc and vcc) repeated 8 times and 64 sounds with no variable aspects (ccc) repeated twice. There is a similar distribution of double source sounds - a greater variety of constant sounds, and more repetition of variable sounds. So a sound like a fog horn, with constant pitch, timbre and amplitude, will not be repeated more that twice; but a completely variable sound - like the ambience in an outdoor market - can be repeated many times. With this repetition of more variable sounds, 1024 sounds can be chosen with only 222 source sounds.

This method must have expanded before the original realization of *Williams Mix*. The score calls for sounds that are not in the 222 sound cards. Also, the collection of the Barrons' tapes for *Williams Mix* in the David Tudor archive [3] contains nearly three times the number of source sounds specified in the deck (although with the same distribution). One possible answer for this divergence is that sounds are replaced with new sounds after being selected from the deck when the score is "mobile". A full analysis of the original Williams Mix tapes and comparison to the score is needed to verify this.

For my version of Williams Mix, I asked a group of my friends to help collect the sounds for the sound library. I did not want the library to be the reflection of a single aesthetic, but rather an aggregate of many people's judgments and methods of sound collection.

Cage suggested processing sounds through filters or reverb to add variant sounds [6] (which might account for the many cricket sounds in the original). I applied 24 different sound processing treatments, with eight affecting frequency, overtone structure and amplitude respectively. These were selected by throwing the I Ching. By processing existing sounds I was able to both increase the size of the sound library and adjust its proportions to match the constant to variable ratio given in Cage's notes. By keeping careful watch on the size, categorization and proportion of aspects in the sound library, I feel I have achieved the same density of variation as is in the original while maintaining the structure of the piece.

PROCESSING TREATMENTS

FREQUENCY
1) Varispeed
2) Chorusing
3) Pitch Shift
4) Ring Modulation
5) Granular Pitch Shift
6) Frequency Shift (SSB Ring Modulation)
7) Phonogene Pitch Shift (Eurorack Module)
8) Brassage (Large Window Granular) Pitch Shift

OVERTONE STRUCTURE
1) Phasor (Swept Notch Filters)
2) Bandpass Filter
3) Spectral Compression
4) Spectral Expansion
5) Convolution
6) Delay
7) Phase Nulling
8) Granular Delay

AMPLITUDE
1) Tremolo
2) Time Domain Brassage
3) Gating
4) Bit-depth Truncation (Decimation)
5) Chebyshev Polynomial Wave-shaping
6) Compression
7) Analog Clipping (Fuzz Factory Guitar Pedal)
8) Granular Sample Playback

Figure 3. Sound library processing treatments for new realization.

4. COMPUTER REALIZATION

The performance software strictly follows the score, and allows only minimal performer interaction during the piece. However, it does not result in a fixed realization, but a new version each time. Cage left much room for variation in the score, and I chose to implement chance procedures whenever a choice is needed. In the original there was similar freedom in the creation and categorization of sounds.

"This is a very free way of permitting action and I allow the engineers making the sounds total freedom. I simply give a list of the sounds needed, e.g.Evcv Fvvv (double source). If a source is ccc by nature, then v means a control. I do not specify how a sound shall be interpreted (in this regard) but leave it to the engineers". [6]

And although he was working with fixed media and creating a single concrete realization, Cage did not want a fixed, repeatable performance of the piece. *Williams Mix* was created for eight tape machines, and the lack of synchronization between machines allowed some variation. Cage states "...my idea all along was to have each track be individual, so that the relation of the tracks could be independent of one another, rather than fixed in a particular scorelike situation." [2]

4.1 Playback engine

The database created from measuring all of the splices was exported to a text file, with a splice on each line, and parameters separated by spaces. Each line is read by the *textfile* object, and scheduled up to 50 milliseconds before it's playtime. When a given splice is played, the next splice is read from the score and scheduled. As there can be up to 8 tracks active at any given time, and splices often overlap, 16 separate splice playback voices are needed to play the sounds.

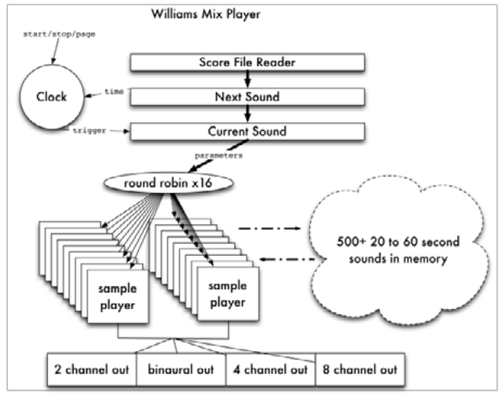


Figure 4. Overall diagram of *Williams Mix* player.

Each splice playback voice is designed to playback any of the splices in the piece, with all of the variations and processing separated into parameters. These parameters include: the channel number, sound category or/ categories (for double sounds), looping, fade shape, and crosscut angle (or lack or crosscut).

Sound selection is rather straightforward, there are around 500 sound files, and the correct one or two files need to be selected for every splice. As there are more sound files than there are categories and aspects, I am able to randomly select the file. If the next sound is a DvcvFvvc, my program will pick one of the 15 Dvcv sounds and one of the 5 Fvvc sounds that I have collected. When Cage adds a number to the sound, like the A1ccc on page 5 of the score, I select the first Accc in the sound file library. Thus there is a mix of fixed and indeterminate sound selection. Cage did not designate in the score what part of a given sound should be selected. As the sounds in the library range from 8 seconds to 60 seconds, and the splice lengths range from 0.01 to 1.66 seconds, I am also able to pick a random start time within each sound file for every splice.

4.2 Looping, crosscutting and pulverization

In the original *Williams Mix*, the tape loops were assembled and transferred to linear tape before they were edited into the master tapes. These loops were typically 1/2 to 2 seconds long in the original sound library. As the splice length in the score is often shorter than the loop length, the repetition that Cage probably expected was usually not heard. In my version, I chose to determine the loop length as 1/2 to 1/12 of the note length, picked at performance time. For very short notes this can produce a ring modulated effect.

The fades are also rather straightforward. In the case of a fade being designated as indeterminate with the horizontal dash in the score, a random fade length is chosen from 0 to twice the length of the splice. When the fade is longer than the splice length, the sound never comes up to full volume. Cage also suggests in the case of indeterminate fades that anything can be done in the editing, even "pulverization". Although I didn't implement "pulverization", I could add a type of time domain randomization to a future version of the piece.

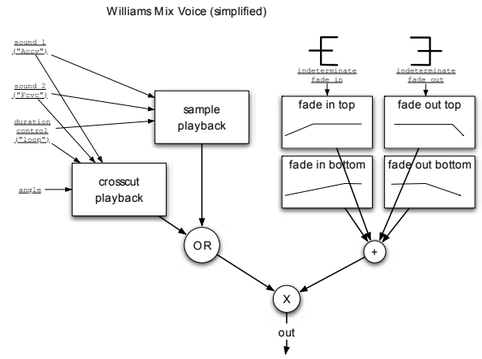


Figure 5. Simplified diagram of *Williams Mix* voice.

The crosscut splicing is one of the more interesting techniques Cage created for *Williams Mix*. Cutting the tape on an angle, rotating the tape, and reassembling has two effects. First, it has the potential of reordering the tape. This early tape technique is very similar to granular sample playback, complete with overlapped trapezoid ampli-

tude envelopes (from the rhomboid shape of the tape splice), and grains that are as short as 20 milliseconds. Second, playing the tape at a respliced angle causes low pass filtering. The angled tape produces the averaged amplitudes for a section of time; similar to a boxcar FIR lowpass filter. These two effects produce one of the characteristic sounds of Williams Mix, a rumbling, granular, bass-heavy timbre. This is implemented in PD with typical granular playback, and low-pass filtering calibrated to match the response of angled tape.

4.3 Spatialization and Performance

Williams Mix is to be performed by eight loudspeakers, surrounding the audience. I have created an output patch which will allow me to use any number of loudspeakers from 2 to 8, mapping virtual loudspeakers using vector based amplitude panning. The virtual speaker positions can be brought to more central locations, for a more monaural playback. This is especially important when in less ideal or more reverberant concert halls. A binaural output is also available.

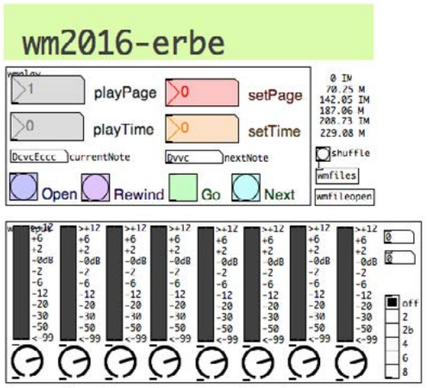


Figure 6. Pure Data main control patch.

The performance controls are simple: *start*, *overall volume*, *individual volume* and *stop*. There is an additional *shuffle* control that randomly reloads the sound library during the performance of the piece. This is only allowed during the sections of the piece when the score is "mobile", and highlights the structure by introducing a new set of sounds to be played and repeated.

5. CONCLUSIONS

Since the competition of the software, I have performed *Williams Mix* nine times, and have released one performance on CD. The most notable difference between the original piece and my version is that of audio fidelity, and possibly the higher fidelity is more capable of making audible the rhythms, structure, spatialization and diversity of material that exists in Cage's score. In retrospect, I do feel that I have been successful in creating a faithful and rich version of Cage's piece. And this realization shows that *Williams Mix* is fully described in the score. Hopefully my score data and other findings can open the way to other researchers and musicians.

I have placed the software and score data for *Williams Mix* at http://tre.ucsd.edu. I have omitted the sound library, leaving the collecting of sounds to the enterprising musician(s).

Several of my versions of *Williams Mix* can be heard at https://soundcloud.com/tomerbe/sets/williams-mix and on the recording *CLPPNG* by Los Angeles experimental hip-hop group clipping. [7].

Acknowledgments

I would like to thank Laura Kuhn of the John Cage Foundation for giving me access to the score, and Jonathan Hiam of the New York Public Library for providing a copy. I would like to thank Larry Polansky, Amy Beal, Anthony Burr, Michael Trigilio, Volker Straebel, and Elizabeth Edwards for comments, critique and research assistance; and Miller Puckette for guidance with Pure Data. Finally, I would like to thank my many collaborators in creating the sound library: Cooper Baker, Bobby Bray, Clay Chaplin, Kent Clelland, Greg Davis, Daveed Diggs, Greg Dixon, Tom Djil, Samuel Dunscombe, Christopher Fleeger, William Hutson, Jeff Kaiser, Scot Gresham-Lancaster, J Lesser, Elainie Lilllios, Stephan Mathieu, Rick Nance, Maggi Payne, Margaret Schedel, Jonathan Snipes, Carl Stone, Michael Trigilio and Doug Van Nort.

6. REFERENCES

[1] J. Cage. *Williams Mix*. 1952. Edition Peters, 1962.

[2] R. Kostelanetz, *Conversing With Cage*. Limelight Editions, 1987.

[3] The Getty Research Institute. The David Tudor papers. Audio Recordings.

[4] J. Pritchett, "David Tudor as Composer/Performer in Cage's *Variations II*", Getty Research Institute Symposium, "The Art of David Tudor", 2001.

[5] J. Pritchett, "The development of chance techniques in the music of John Cage, 1950-1956", Ph. D. Thesis, New York University, New York, 1988.

[6] J. Nattiez, *The Boulez-Cage Correspondence*. Cambridge University Press. 1993.

[7] Clipping., *Clppng*, Sub Pop Records, 2014.

[8] L. Austin, "John Cage's *Williams Mix (1951-3):* the restoration and new realisations of and variations on the first octophonic, surround-sound tape composition," *A Handbook to Twentieth-Century Musical Sketches*. Cambridge University Press. 2004.

[9] V. Tricoli, W. Dafeldecker, *Williams Mix Extended. http://www.dafeldecker.net/projects/pdf/WME_Audio%20copy.pdf* . 2011.

Computer-Based Tutoring for Conducting Students

Andrea Salgian
Department of Computer Science
The College of New Jersey
Ewing, NJ, USA
salgian@tcnj.edu

David Vickerman
Department of Music
The College of New Jersey
Ewing, NJ, USA
vickermd@tcnj.edu

ABSTRACT

In this paper we present a computer-based conductor tutoring system that uses the Microsoft Kinect to provide beginner conducting students with feedback about their performance during an individual practice session. The system is capable of detecting common mistakes such as swaying, rocking, excessive hinge movement, and mirroring, and it can also determine conducting tempo, as well as classify articulation as staccato or legato. Testing has shown that the systems performs nearly perfectly when detecting rocking, swaying, and excessive hinge movement, correctly classifies articulation most of the time, and determines tempo correctly. The system was well received by conducting students and their instructor, as it allows them to practice by themselves, without an orchestra.

1. INTRODUCTION

One of the enduring challenges facing teachers of conducting is the lack of immediate feedback available to their students while practicing. Student musicians who have conventional skills in pitch and rhythm can usually immediately detect and improve wrong notes and rhythms due to the real-time audio feedback from the sound itself. However, conducting students might receive instruction and feedback during class, but when attempting to practice these newly learned physical gestures, they are without a responsive ensemble, and thus unable to receive immediate feedback on the effectiveness of their gestures. The Conducting Tutor project, using the Microsoft Kinect camera, aims to provide a solution to this problem by providing a tool that students can use in order to receive immediate visual feedback on the effectiveness of their conducting gestures. Our system is small and simple enough that a student can set it up and use it in her room.

Many recent musical interaction systems use computer vision to allow a musician, or even the general public, to conduct a virtual orchestra [1, 2, 3]. But pure vision-based systems may have difficulty in tracking the conducting baton or hand. Better performance was obtained by systems using batons equipped with sensors and/or emitters, such as the Digital Baton system implemented by Marrin and Paradiso [4], and the *Virtual Maestro* by Nakra *et al*. [5].

More recent systems are aimed at conducting real orchestras [6], or educational purposes [7, 8].

In this paper we describe a system that uses the Microsoft Kinect to track the hand gestures of a student conductor, and provides real time feedback about leaning, swaying, mirroring, as well as conducted tempo and articulation. The system was tested on several students with good results.

2. SYSTEM OVERVIEW

Early in the project it was critical to determine which elements of physical conducting could actually be tracked, analyzed, and displayed by the camera. We realized that the camera could not catch delicate, small gestures, so it was determined that the tutor program could be most helpful to beginning conducting students whose gestures and posture issues would be more likely to be tracked by the camera. (As students advance in their understanding of conducting gestures they are routinely encouraged to make some gestures smaller and it was discovered that the Conducting Tutor was less likely to discern articulation pattern changes in these small gestures.) The skills that the Conducting Tutor aims to improve include some of the essential physical movements and posture issues that are encountered in basic conducting courses: swaying and leaning, hand independence and basic articulation styles (legato and staccato).

Many beginning conducting students tend to sway from side to side or lean forward or back, which can be distracting to an ensemble and subsequently weaken the effectiveness of the conductor's gestures. The same goes for conductors who constantly mirror gestures in both arms. It is a desired goal of most conducting pedagogies that a student be able to demonstrate independent use of both the right and left hand/arm.

One of the more difficult elements for beginning students to self-assess is whether the style/articulation of their gestures matches the music they are trying to communicate. The primary difference between these two articulations in gesture is the speed at which the ictus is approached and performed. In conducting, the ictus is the moment the beat occurs. A musical passage that is predominantly staccato requires quick, sharp movement to the ictus, while a passage that is primarily legato requires a wider, softer movement to the ictus.

Conductors traditionally practice in front of a mirror. Our system provides the "mirror" by showing the live video of the user on the screen, and it augments it with information displayed as color and text on the right side of the screen (see Fig. 1). This information includes the amount of time swaying, rocking, and excessive hinge movement occurs, and it is displayed as a percentage of the total conducting time. If any one of these mistakes is currently occurring, the indicator color changes from

green to red. Should the percentage go above a preset threshold, the color of the text box changes from green to red.

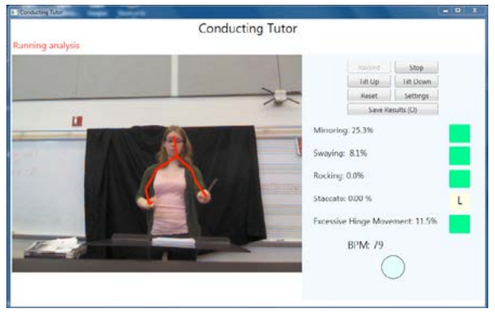


Figure 1. Screen capture of the conducting tutoring system.

The current articulation is also displayed on the screen as an L on yellow background for legato, and an S on blue background for staccato, together with the percentage of time spent conducting staccato.

Finally, the system estimates the number of beats per minute (bpm) conducted by the user, and displays it on the screen.

3. METHODOLOGY

The Microsoft Kinect is a motion sensing device that enables gesture recognition by detecting the skeleton of a human figure and tracking its joints. Tracking can happen in one of two modes: standing mode and seated mode. Standing mode tracks twenty joints, while seated mode tracks only the ten joints in the upper half of the body (shoulders, elbows, wrists, arms and head). Often, conductors have a music stand in front of them, which obstructs the view of their lower body. Since this can lead to unpredictable system behavior, we opted to use the seated mode instead of the default standing mode. Note that use of this mode does not require the user to be seated (which would be an awkward position for conducting); it merely means that the Kinect tracks only the upper body skeleton.

Since conducting technique is best demonstrated by example, we started by asking a conducting educator and an advanced conducting student to demonstrate correct conducting of a number of musical pieces with varying tempo and articulation, as well as isolated incorrect techniques in a continuous fashion (e.g. continual swaying, continually excessive hinge movement). We recorded these performances and analyzed the motion of the skeletal points as tracked by the Microsoft Kinect.

3.1 Detection of Swaying, Rocking, and Excessive Hinge Movement

Graphing coordinates for relevant joints can easily show where and by how much normal and erroneous movement differ. We picked the "center shoulder" joint, located at the base of the neck, to detect swaying. In seated mode, the spine and other lower joints are not tracked, so we can only identify swaying based on the amount of change we see in the x-coordinates of this joint. The base of the neck does not move in relation to the rest of the body, and it is centralized and infrequently obstructed during conducting.

Fig. 2 shows a time series plot of the x-coordinate of this joint for correct and incorrect techniques. As we can see, with correct conducting technique, the conductor does not sway or move from their central location at all. The slight noise is easily accounted for with a small threshold. On the other hand, there is significant movement when the conductor is swaying. By identifying a threshold and a general number of frames it takes to exceed this threshold when the conductor moves, we were able to identify swaying movements using only the x-coordinates.

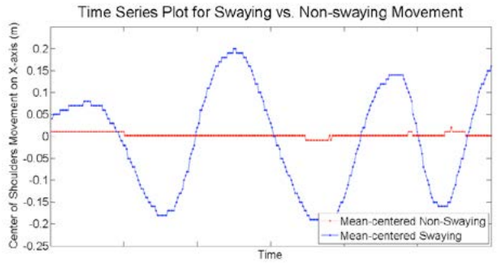


Figure 2. Mean-centered plot of center shoulder coordinate during swaying and non-swaying motions.

Detection of rocking forward and backward works similarly.

Excessive hinge movement primarily involves swinging of the elbows, which, like the conductor's body, should remain relatively stationary. Since incorrect motion happens along all three dimensions, our algorithm calculates the Euclidean distance between consecutive elbow locations over a certain number of frames. A distance larger than a threshold is classified as incorrect conducting technique.

3.2 Mirroring Detection

Mirroring is detected somewhat similarly to swaying and rocking. This time we need to analyze the relative position of two joints that are not stationary, the two hands, over time, by looking at their x, y, and z coordinates. Fig. 3 shows two plots: on top, the x-coordinates of the hands, mirrored over the center of the body using the center shoulder joint; on the bottom, the y-coordinates of both hands. In both plots, the right hand, which remains on a steady beat, is shown in red. The left hand when mirroring constantly is shown in pink. It is clear that when mirroring, the locations of the two hands is similar, though not perfectly symmetrical.

When mirroring, the left hand mimics the right hand very closely in both dimensions. The blue line in both graphs shows the proper technique. While similar to the right hand in some places, usually the left hand is making some other motion and is not conducting the beat like the right hand. Near the end of the time shown, the conductor is mirroring for about one beat. However, instead of mirroring the entire time, the conductor also cues with the left hand for entrances or cutoffs.

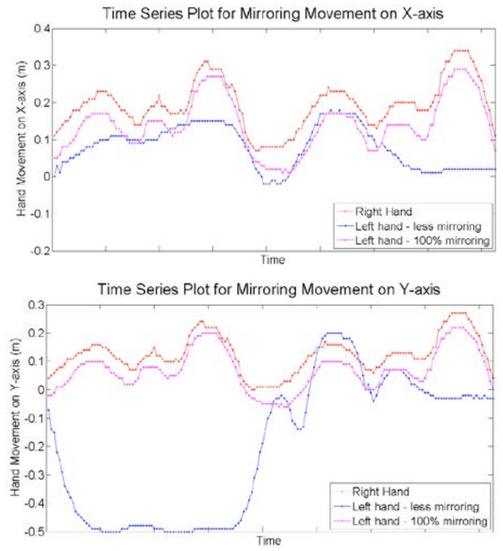


Figure 3. X-coordinates (top) and Y-coordinates (bottom) of hands during mirroring.

Since mirroring of the hands is not exclusive to incorrect technique, the user can adjust the threshold at which mirroring is considered incorrect.

3.3 Tempo Calculation

Our algorithm uses the right hand coordinates provided by the Kinect tracker to compute the instantaneous velocity and looks at neighboring instances to detect changes in direction. Each change in direction is counted as a beat. The timing of the ten most recent beats is averaged to extrapolate the beat per minute (bpm) count, which is then displayed on the screen.

3.4 Recognizing Articulation

Of all the conducting characteristics analyzed by our system, articulation was the hardest to quantify. The difference between conducting legato and staccato seems obvious, yet one has difficulty describing it in words. The sharpness of staccato movements seemed to suggest that the difference would lie in the velocity or acceleration of the right hand. We used the hand coordinates obtained by the Kinect to compute instantaneous and average velocities and accelerations, and we noticed that the acceleration was the same for staccato and legato pieces. If the tempo was the same, the average velocity was also the same regardless of the articulation.

We have found that due to the tempo patterns, the right hand velocity magnitude varies constantly, peaking right before the hand changes direction. The difference between staccato and legato lies in the height of these peaks, staccato gestures speed up and slow down significantly more than legato gestures. This can be seen in Fig. 4, which shows how the velocity magnitude varies over time. The same musical piece was conducted in staccato, with the velocity shown in blue, and in legato, with the velocity shown in orange. The velocity magnitude had to be smoothed due to occasional tracking errors by the Kinect.

Given this finding, our algorithm computes the peak height in velocity magnitude and uses a threshold to distinguish between legato and staccato. Since conducting styles vary widely, this threshold can be adjusted by the user.

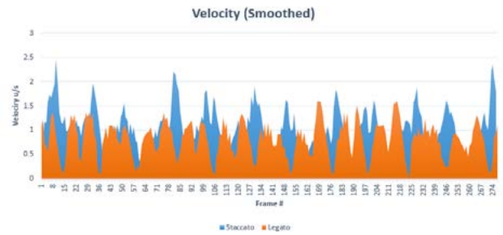


Figure 4. Smoothed magnitude of velocity of right hand conducting a staccato piece (blue) and a legato piece (orange). Staccato is characterized by peaks of higher magnitude.

4. RESULTS

The system was tested on ten students who conducted the introductory part (called Theme) of Edward Elgar's Enigma Variations as part of their assessment in the Conducting I course at our institution. The piece is characterized by varying tempo, making conducting (and assessment) more difficult. The articulation also changes, starting out as legato and switching to staccato in the middle. To make sure testing was not biased, students could not see the computer screen, and received feedback only from the Conducting instructor. The instructor could not see the computer screen either, thus his decisions were independent of the program output. We recorded the program output, as well as the instructor's feedback, and compared the two.

The students conducted anywhere between 40 and 100 seconds (until they were stopped by the instructor), always starting from the beginning of the piece.

We found that the mirroring, as well as swaying, rocking, and excessive hinge movement was always correctly detected.

Articulation was correctly classified 70% of the time.

Tempo measurement performance varies because of the Kinect framerate.

We also tested the system allowing students to see the instant feedback on the screen. Both the students and their instructor found the system very helpful and easy to use.

5. CONCLUSIONS AND FUTURE WORK

In this paper we described a computer-based system that uses the Microsoft Kinect to provide real-time feedback about conducting performance.

Our method uses the trajectory of upper body joint coordinates to detect swaying, rocking, excessive hinge movement, and mirroring, common mistakes made by novice conductors. We perform beat per minute calculation to determine tempo by looking at abrupt changes in the direction of right hand motion. Finally we classify articulation as legato or staccato by looking at how the magnitude of right hand velocity changes over time.

The system provides the user with a mirror image and overlaid upper body skeleton as tracked by the Kinect, together with instantaneous analysis of the current gesture and overall performance statistics.

We tested the system on several students conducting a difficult musical piece containing changes in tempo and articulation, and we have found that incorrect gestures are always detected. Articulation is correctly classified 70% of the time, while tempo calculation still has room for improvement.

The system was very well received by users, as it fills an important need of feedback for novice conductors that practice by themselves without an orchestra or an instructor.

Future work includes refinement of the tempo calculation method, extensive rigorous testing, and the addition of more conducting technique elements.

Acknowledgments

The authors would like to thank students Leighanne Hsu and Nate Milkosky who contributed to this project by writing code and testing.

6. REFERENCES

[1] R. Behringer, "Conducting Digitally Stored Music by Computer Vision Tracking", *Proceedings of the First International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution* (AXMEDIS'05), Florence, Italy (2005)

[2] A. Wilson, A. Bobick, "Realtime online adaptive gesture recognition", *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems,* Corfu, Greece (1999)

[3] D. Murphy, T. H. Andersen, K. Jensen, "Conducting audiofiles via computer vision", *Proceedings of the 5th International Gesture Workshop*, LNAI, Genoa, Italy (2003) 529-540.

[4] T. Marrin, J. Paradiso, "The digital baton: A versatile performance instrument", *Proceedings of the International Computer Music Conference,* Thessaloniki, Greece (1997) 313-316

[5] T. M. Nakra, Y. Ivanov, P. Smaragdis, C. Ault, "The UBS Virtual Maestro: an Interactive Conducting System". *New Interfaces for Musical Expression (NIME),* Pittsburgh, PA (2009)

[6] A. Salgian, M. Pfirrmann, T. M. Nakra, "Follow the Beat? Understanding Conducting Gestures from Video", *Proceedings of the International Symposium on Visual Computing (ISVC).* Lecture Notes in Computer Science. (2007).

[7] L. Peng, D. Gerhard, "A Wii-based gestural interface for computer-based conducting systems", *New Interfaces for Musical Expression (NIME),* Pittsburgh, PA (2009)

[8] E. Ivanova, L. Wang, Y. Fu, J. Gadzala, "MAESTRO: a practice system to track, record, and observe for novice orchestral conductors", *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. Pages 203-208.

[9] MSDN Library. Kinect for Windows SDK. https://msdn.microsoft.com/en-us/library/hh855347.aspx.

a Band is Born: a digital learning game for Max/MSP

Oliver Hancock
Nelson Marlborough
Institute of Technology
oliverjhancock@googlemail.com

ABSTRACT

a Band is Born is a musicians' introduction to programming Max/MSP in the form of a digital learning game. The game world is implemented directly in Max's patch editor. The pedagogical underpinnings of the game in a learning context are presented. The game is briefly described, and evaluated using the Sig-Glue framework and anecdotal evidence. It is concluded that Max lends itself to construction play and constructivist learning, and that a Band is Born promotes the play-based learning traits of persistence and engagement which are desirable in effective novice programmers.

1. INTRODUCTION

a Band is Born is a digital learning game designed to introduce musicians to Max programming. It uses Max's own patch editor as its platform rather than Jitter. Graphics are restricted to straightforward picture display and simple animations. Gameplay consists of programming with Max's patcher in the normal way.

There exist a wide range of materials and resources for learning Max/MSP. These include the software's native tutorials and help files [1], books [2], video tutorials [3] and complete course materials [4]. As far as the author is aware, this is the first digital learning game which provides a sizeable resource, covering basic sound generation, DSP, control (including external hardware controllers) and sequencing across synthesis, sampling and live audio.

The primary learning outcome is to instill confidence and a working engagement with Max; specific and accurate understanding of Max objects is secondary, as is knowledge of programming as a discipline. The game can also provide resources for students' further work.

2. PEDAGOGY

The approach is constructionist: learners actively build or modify Max patches, forming their own understandings as they work [5]. *Construction play*, even purely for enjoyment, nevertheless involves *epistemic* (learning) and *ludic* (playing) behaviours [6], with the former being further classified into *differentiation* (understanding individual components) and *integration* (conceptualizing constructions of many components) [7]. Play-based learning can promote relaxation, motivation, persistence, concentration, exploration of new skills and creativity [8]. Persistence and experimentation are cited as effective behaviours for novice programmers [9].

However, constructionist understandings can be precocious or flawed [10], and novice programmers generally are found to suffer from 'fragile' knowledge of their programming language and strategies [11].

Practical teaching approaches will ideally include non-gaming activities. The VISOLE pedagogy [12] 'encompasses scaffolding, online game-play, and teacher-facilitated debriefing and reflection.' Problem-based learning (PBL) is also a suitable model providing minimal instruction and coaching as scaffolding which is reduced as learners progress [13]. PBL may be considered a close parallel to the programming process itself, as described in [14, 15, 16]. *a Band is Born* includes quiz zones which check knowledge before players can move to higher levels.

The game structure provides multiple pathways to learn Max, offering variation of content, structure and pace. Players may navigate freely between pathways and levels. The material has high redundancy and players need not visit every zone of the game in order to achieve its gaming challenge, or its learning outcomes.

3. THE GAME

The scenario is an imaginary city's CBD. The opening screen defines the game task: this is a "build-a-band" game (Figure 1).

The game has an *open world*, with players able to freely navigate among 150 zones (rooms within the various shops and buildings), where they find characters representing professional and amateur musicians. They also encounter ready-made Max patches and partially built ones with instructions for completing them (Figure 2). Less explicitly identified actions are possible: for example clicking on café owner Bob's badge reveals that he is a drummer; and his toilet can be visited to discover a sample playback patch.

The graphics are in first-person, isometric view. They are pre-rendered but dynamic, using pictctrl and pictslider for basic animation; and of course Max's own GUI is activated in the patch-building play. The zones are nested as sub-patches allowing players to move between zones without leaving the game environment.

Figure 1. The title screen of a Band is Born.

Figure 2. The Café zone.

There are limited sound effects, and these fade rapidly at the start of a new zone, like an aural cutscene, so as not to obscure the sounds from the patch once it is built. All musical material is initially presented on a C major chord at 120bpm, affording modularity: sounds, riffs, beats and chord patterns can be combined freely (but without aesthetic guarantees).

The language of dialogues is non-technical where possible to give the game a chatty style. This is a resource primarily for musicians rather than programmers, and aimed at developing a working knowledge. The patches favour simplicity and clarity rather than elaborate functionality, with many extraneous number objects and buttons added for monitoring.

To reduce scaffolded learning, the graphics associated with the game world gradually give way to a more conventional Max patcher appearance as players move further into the levels and zones. *a Band is Born* was originally developed to harmonize with the visual style of Max 6, but runs in Max 7. It is not fully compatible with Max 5.

All building exteriors and interiors are modeled using pCon.planner 6, a simple to use, free software which can also import 3D models made in other programs or from an extensive library. Characters are created from photographs low-resolution treated in Adobe Illustrator CC to help with editing, compositing, and with stylistic consistency across the game. Some elements are drawn in MS Powerpoint and Tinkercad. Final compositing is completed in Power Point, including its clipart.

This production method enables teachers to create new zones matching the existing style, and they are encouraged to do so in the copyright notice. There are also empty or 'template' zones in the game to allow for new material.

4. EVALUATION

4.1 Sig-Glue

Earlier play-based learning resources for Pure Data [17] were evaluated using the Sig-Glue framework [18]. That has also been used for *a Band is Born* to facilitate comparison. The framework incorporates ideas from several other approaches but remains wieldy, and can be applied to finished resources without the need for play testing or user feedback.

Evaluation consists of noting the presence or absence of various features. Because the list is long, a summary is presented here showing the proportions of features present and absent, as well as those present with qualifications or not applicable. *a Band is Born* scores better than the earlier Pure Data resources.

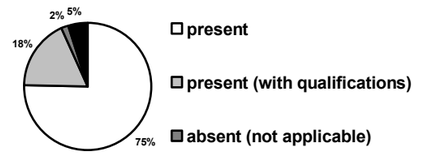


Figure 3. Sig_Glue evaluation of *a Band is Born*.

4.2 In Practice

To date this resource has only been used in brief sessions within elective courses designed to make music with ready-made and custom-built controllers. Users were music undergraduates with no prior programming experience, although most were familiar with DAWs and some had used controllers such as Novation Launchpad. The session using *a Band is Born* was a two hour crash-course intended to give students hands on experience with Max, and the confidence to work with scaffolded patches to complete their controller-based projects.

With such a short period of gaming under busy workshop conditions, and with Max learning forming just a small part of the overall course, the following observations must be regarded as anecdotal evidence.

Participating voluntarily in the course, the students had minimal apprehension about using music software in general or even programming. Nevertheless their initial engagement was perhaps greater than might be expected with a traditional lecture format, or even self-directed learning in a non-game format. Having said that, it seemed that they were aware of the need to learn quickly and most did not actually take time to enjoy the purely ludic features of the game.

Students were encouraged to visit the Youth Centre zone which includes patches handling hardware games controllers. One student expressed the opinion that more of this type of patch would have been helpful within the specific controllers- course context. This suggests that for differing course requirements, targeted variations of the basic game could be prepared.

One student took up Pure Data programming after the Max-based course, and reported that transferring was trouble free. It seems that *a Band is Born* could be used as an introduction to Pure Data; perhaps it might be tweaked to make more use of objects which the two languages have in common. There is no need to save anything in gameplay, so a free demonstration copy of Max could be used for learning.

Compared to the author's earlier play-based resources for Pure Data and Max [17] (which used multiple stand-alone patches structured within folders in the Macintosh finder window) this immersive game world did not sustain interest for so long. Students suggested that one to one-and-a-half hours would have been enough. *a Band is Born* lacks the variety of the earlier resources in both themes and visual style. Perhaps this induces boredom sooner. Alternatively the constant immersion in the game world may lack the brief rests and change of activity associated with finding a new patch from the finder window. A more positive suggestion is that students felt ready to work outside the game sooner using *a Band is Born* than with the earlier, more diverse resources. Certainly some students had begun to develop their own patches within the time allotted for play. It is hoped that with a more leisurely timeframe for learning, students would enjoy multiple, short periods of game-play and thus persevere with the game for a longer period.

5. CONCLUSIONS

Early indications are that *a Band is Born* works as expected when used as a digital learning game. The close alignment between Max's visual programming environment, construction play, play-based learning and the actual task of programming are borne out in practice. Learning with this resource and then progressing to novel programming seems coherent, natural and even seamless, especially when supported by reducing scaffolding. Students rapidly and enthusiastically engage, and soon exhibit the persistence and experimentation identified as desirable behaviours for effective programming novices.

However, also as expected, constructivist knowledge of Max can be inconsistent and inaccurate. This exacerbates the fragility of knowledge generally

found in novice programmers. Nevertheless, students who remain engaged and who continue programming have at least a chance to correct misunderstandings.

6. FUTURE WORK

This game is in the early stages of development. The author invites colleagues to use the resource, alter it and provide feedback.

The possibilities of varying the game to suit different courses and perhaps different age groups of learners seem worthy of exploration. That also raises questions about the optimal size of the game world; could it be released in a modular format? What expectations are realistic for using *a Band is Born*? Can it be a complete introductory course for Max; or is it most useful as an icebreaker to be followed by other modes of instruction; or is a sustained combination approach such as VISOLE more effective?

What extra resources could accompany the game? Would players benefit from hard copy materials, perhaps in the form of a game strategy guide? (This might be an appropriate way to incorporate instructionist material within the broader constructionist pedagogy). Should the game contain a link or directions to the standard Max tutorials? What extra resources would teachers appreciate: component images forming the graphics; suggested uses and projects; course outlines; session plans; lecture notes; visual aids; student worksheets; questionnaires?

a Band is Born is available to download free at: *https://sites.google.com/site/oliverjhancock/*

7. REFERENCES

[1] Cycling '74 "Start patching now with Max 7.1". Internet: https://cycling74.com/downloads/#.VtOVzFt9600 [29/2/2016].

[2] A. Cipriani and M. Giri, *Electronic Music and Sound Design*, ConTempoNet 2010.

[3] H. Jackson, "MaxMSP Tutorial 1- The very basics". Internet: https://www.youtube.com/watch?v=nDBbxRvVwxw [29/1/2016].

[4] M. Phillips, "MaxMSP-based Teaching and Learning Resources". Internet: http://www.ohio.edu/people/phillipm/public_html/MaxResources.html [29/2/2016].

[5] S. Papert, *Mindstorms*, New York, Basic Books, 1980.

[6] S. J. Hutt, S. Tyler, C. Hutt and H. Christopherson, *Play, Exploration and Learning: a Natural History of the Pre-School*, London, Routledge, 1989.

[7] P. Gura, "Developmental Aspects of Blockplay", in P. Gura (ed.), *Exploring Learning: Young Children and Blockplay*, London, Paul Chapman, 1992.

[8] M. Prensky, *Digital Game-Based Learning*, New York, McGraw-Hill, 2001.

[9] D.N. Perkins, C. Hancock, R. Hobbs, F. Martin, and R. Simmons, "Conditions of learning in novice programmers," in E. Soloway and J.C. Spohrer (eds.), *Studying the novice programmer* (pp. 261–279). Hillsdale NJ, Lawrence Erlbaum, 1989.

[10] R. Spiro and M. De Schryver, "Constructivism When It's the Wrong Idea and When It's the Only Idea," in S. Tobias, and T. Duffy (eds.), *Constructivist Instruction Success or Failure?,* New York, Routledge, 2009.

[11] D.N. Perkins and F. Martin, "Fragile knowledge and neglected strategies in novice programmers," in E. Soloway and S. Iyengar (eds.), *Empirical studies of programmers, First Workshop* (pp. 213–229). Norwood NJ, Ablex, 1986

[12] K. Cheung, M. Jong, F.L. Lee, J. Lee, E. Luk, J. Shang, and M. Wong, "FARMTASIA: an online game-based learning environment based on the VISOLE pedagogy," *Virtual Reality* 12, pp. 17-25, 2008.

[13] C. Hmelo-Silver, "Problem-Based Learning: What and How Do Students Learn?" *Educational Psychology Review* 16(3), pp. 235-266, 2004.

[14] S.P. Davies, "Models and theories of programming strategy," *International Journal of Man-Machine Studies* 39, pp. 237-267, 1993.

[15] T.R.G. Green, "Programming languages as information structures," in J.M. Hoc, T.R.G. Green, R. Samurcay, and D.J. Gillmore (eds.), *Psychology of programming* (pp. 117–137). London: Academic Press, 1990.

[16] W. Visser, "More or less following a plan during design: Opportunistic deviations in specification," *International Journal of Man-Machine Studies* 33, 247-278, 1990.

[17] O. Hancock, "Play-based, constructionist learning of Pure data – a case study," *Journal of Music, Technology and Education* 7(1), pp 93-112, 2014

[18] C. Dondi, and M. Moretti, "A methodological proposal for learning games selection and quality assessment," *British Journal of Educational Technology* 38(3), pp. 502-512, 2007.

Detecting Pianist Hand Posture Mistakes for Virtual Piano Tutoring

David Johnson
University of Victoria
davidjo@uvic.ca

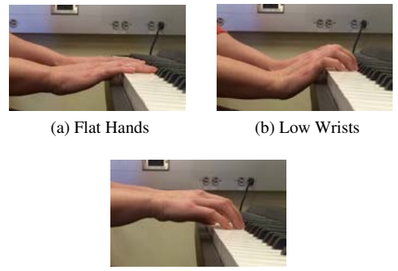
Isabelle Dufour
University of Victoria
idufour@uvic.ca

Daniela Damian
University of Victoria
gtzan@uvic.ca

George Tzanetakis
University of Victoria
danielad@uvic.ca

ABSTRACT

Incorrect hand posture is known to cause fatigue and hand injuries in pianists of all levels. Our research is intended to reduce these problems through new methods of providing direct feedback to piano students during their daily practice. This paper presents an approach to detect hand posture in RGB-D recordings of pianists' hands while practicing for use in a virtual music tutor. We do so through image processing and machine learning. To test this approach we collect data by recording the hands of two pianists during standard piano exercises. Preliminary results show the effectiveness of our methods.



(a) Flat Hands (b) Low Wrists

(c) Correct

Figure 1. Hand Posture Classes

1. INTRODUCTION

Learning to play piano is a challenging task that requires years of disciplined practice to master. Typically, aspiring pianists rely on weekly lessons with a professional teacher to supervise their learning progress. In order to improve their playing abilities, students must augment weekly lessons with daily practice where they are expected to gradually be able to self-analyze their performance. However, students must wait for each lesson to receive expert feedback on their practice and technique.

Research in Computer Assisted Music Instrument Tutoring (CAMIT) systems attempts to solve this problem by providing the tools necessary to analyze students' performance and provide personalized feedback [1, 2, 3]. Typically the feedback students receive only takes into account the musical quality of the performance, omitting evaluation and feedback based on their posture and technique. Our work intends to fill this gap with a system able to *watch* a student perform daily exercises to provide feedback on hand posture.

1.1 Pianist Hand Posture

Riley et al. [4] discuss the importance of performance feedback in musical skill acquisition, especially in the case of repetitive practice where consistent bad technique may lead to injuries or fatigue. In contrast, our system is intended to provide immediate performance feedback without arduous analysis as presented in [4].

For a correct hand posture, the hand should be arched and the fingers curled as illustrated in Figure 1c. One of the authors (a piano teacher) identified two common posture mistakes observed in students: playing with flat fingers (see Figure 1a), and playing with low wrists (Figure 1b). Because most of the practice time occurs between lessons, bad habits can quickly become chronic. Providing a tool that can identify and help correct these mistakes during daily practice would reduce the probability that they become ingrained in the student's playing style.

To analyze hand posture, Li et al. [5] find key points in the hand, such as the center of the hand, the middle finger, and the wrist. They use the key points to derive two features: the center height-to-hand arch ratio and the wrist angle-vertical. Then, they collect data from expert pianists and use the features to calculate values that indicate deviations from normal posture. We propose a different approach to perform posture analysis by modeling the entire hand using computer vision based features generated from depth maps. These features are then used to train machine learning models for hand posture detection.

1.2 Hand Pose Recognition

In order to identify hand posture, the pianist's hands must be segmented from the depth map. Typically, hand segmentation from depth maps is performed in research related to hand pose recognition. The goal of pose recognition is to infer the pose of a hand from a database, such as sign language digits, which then can be mapped to an ac-

tion [6]. In most cases, the hand is straightforward to segment from depth maps since it is assumed to be the closest object to the camera and is not interfered with by other objects. Pianists' hands, however, are in contact with piano keys and in some cases parts of the hands or wrists may be below the keys. Moreover, the shape of pianos and design of practice spaces may vary.

1.3 Pianist Hand Detection and Pedagogy

Identifying and tracking pianists' hands for pedagogical applications has been explored in previous research. Tits et al. [7] used a marker based motion capture system to analyze pianists' hands and finger gestures to determine the performer's level of expertise. Marker based approaches are generally intrusive and not readily available to non-researchers. As an alternative, markerless approaches for hand tracking use standard RGB cameras [8] or depth maps from RGB-D cameras, such as the Kinect [9, 10, 11]. Hadjakos and Lefebvre-Albaret [8] presented three methods for using RGB video to detect which hand played a note, while Oka and Hashimoto [9] used a combination of depth recordings from a Kinect and information from MIDI data to identify correct piano fingering. While all of these works generate information that can be used in learning applications, none model the shape of the hand for posture detection.

2. SYSTEM DESCRIPTION

In this paper, we propose an approach for detecting the hand posture of a pianist during practice using RGB-D cameras, such as the Kinect or Intel Realsense. The use of a this type of camera affords easy installation in any practice space and a non-invasive setup with the camera located above the hands, as shown in Figure 2. The camera is positioned above the piano to capture both hands with one camera. This placement also affords the opportunity to obtain additional information from recording, such as the keys being played.

Figure 2. Piano and Kinect Setup

VPT is intended for use at the homes, or other practice locations, of students, each of whom may have different hand sizes. Additionally, it may need to be configured by non-technical users, so training a good model should require minimal data and effort. To meet these requirements, we present a method that supports per-user training by non-technical users with a short initial setup time. Similar detection models trained for individual users are already being used in real-world applications like Microsoft's Visual Gesture Builder [1].

2.1 Configuration

This approach requires some initial configuration steps from the users. The first step is to train a background subtraction model by simply recording the practice space without the student for a few seconds. Afterwards the hand posture detection models are trained. To make this step easy for students and teachers, the model is trained with static hand postures. With the help of the teacher, the student holds both hands for ten seconds in a static position for each category of hand posture to be detected. This training scheme affords both minimal annotation and personalized training based on common mistakes that have been observed for a particular student. The following sections discuss the implementation details of the hand posture detection system.

2.2 Hand Segmentation

In our approach, before detecting hand posture the left and right hands must be segmented from the depth map. This is done through a combination of background subtraction and thresholding.

To account for variation in practice spaces, the first step of segmentation is to remove the piano and other static objects from the scene. This is done by generating a foreground mask using Gaussian Mixture Model based background subtraction [12]. Next, morphological opening (i.e. erosion then dilution) is applied to the mask to remove noise objects. The foreground is obtained by applying the generated mask to the original depth map.

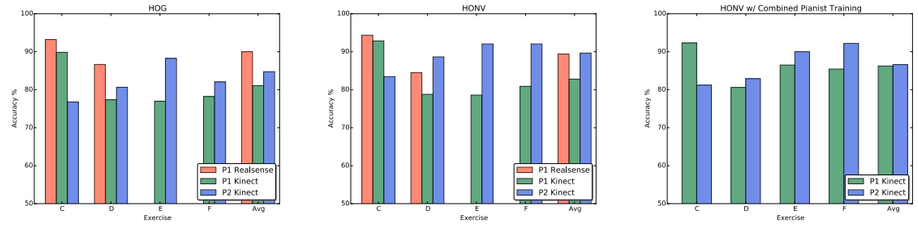
Depending on the range of the camera, additional data such as the pianist's legs may be included in the foreground mask. To remove these additional objects, we take advantage of the fact that the hands will be the closest object to the camera. Since the piano has already been removed form the scene via background subtraction, there is a gap between the hands and thighs (which are the next closest object). Using this observation, thresholding is performed to remove depths greater than the depth at that gap. The thresholding value is obtained by finding the bin at the local minima after the first peak of a depth histogram (not including bin zero). Applying the threshold generates a foreground containing only the hands.

Once the hands have been segmented from the depth data, the location of each hand needs to be identified. The bounding box of each hand is derived through Canny Edge detection followed by a contour analysis of the edges [13]. Due to the orientation of the camera above the piano, the right hand is contained in the bounding box closest to the left edge of the image, i.e. the center of the box with the smallest x value. The final output is the bounding box coordinates for each hand.

2.3 Feature Extraction

Two feature sets are utilized to model the hand for posture detection, Histograms of Oriented Gradients (HOG)

[1] http://goo.gl/qxILqW



(a) (b) (c)

Figure 3. Hand Posture Detection Accuracy Rates

and Histograms of Normal Vectors (HONV). HOG is often utilized as a feature set for object recognition in RGB and grayscale images [14]. The key idea behind HOG is to capture local shape through edge strength and direction. HOG features are calculated by approximating the derivative of color intensity in the X direction and in the Y direction. The X and Y gradients are converted to polar form to generate orientation angles and a magnitudes for each pixel in the image. Histograms are generated for sliding non-overlapping blocks. For each block, orientation angles are voted into bins with the votes weighted by the magnitudes, thus, capturing both the direction and strengths of change. When applied to depth maps these features capture the shape of an object similarly via edge direction but also by capturing the depth gradients on the surface of the object. For example, when a pianist is playing with their wrist too low, the gradients of the top of the hand will be greater than when playing in correct form in which case, the top of the hand is flat. HONV was developed to provide a geometric representation of objects in depth maps [15]. For HONV, the X and Y gradients are used to calculate the azimuth and zenith angles of normal vectors of unit magnitude. Then, the angles for each pixel are voted into two dimensional histograms.

Due to the varying state of the hand while performing, bounding boxes may change in size from frame to frame, which makes it difficult to use the standard block approach in this work. For example, while playing a pianist may need to stretch their fingers to reach keys, thus, making the detected hand image wider than normal. The varying width of the hand means that histograms cannot be calculated using standard blocks. There is much less variability in the length of the hand while playing with a specific hand posture. For example, the wrist is usually the same distance from longest fingertip. To account for the variable hand width, instead of using sliding blocks, histograms for the features of each hand are calculated using horizontal slices of the detected hand image.

3. EXPERIMENTS

To test our approach for hand posture detection, data is collected by recording the performances of two pianists while they play material from lesson plans of beginning piano students. Two depth cameras are tested during data collection, the Microsoft Kinect and the Intel Realsense. Hand posture detection is performed with models for each hand using Support Vector Machines (SVM) with a linear kernels. A one-vs-all strategy is implemented for multiclass classification.

3.1 Data Collection

Data was collected over three recording sessions with two different pianists. Pianist P1 is a piano teacher that plays at an advanced level and Pianist P2 plays at an intermediate level. In the first session, P1 is recorded with the Realsense depth camera. For sessions two and three, the Kinect is used to record P1 and P2 respectively. In each session, the pianist performs the following set of exercises.

During the first session P1 performs four different exercises in each hand posture category. Exercise **A** consists of P1 holding her hands in static pose for each category (correct, flat hands, and low wrists). For exercise **B**, P1's hands are held in a static pose but this time with keys pressed. In exercises **C** and **D**, motion is added for a more realistic dataset: exercise **C** is a C major scale and exercise **D** is a technical exercise from the popular piano lesson book series *A Dozen a Day*. For the second and third sessions, exercises **E** and **F** from the same lesson book are added for each posture category.

3.2 Results

We implement 5-fold cross validation for each recording session to test the performance of our approach. Left hand and right hand datasets are generated for each session using the proposed segmentation and feature extraction process. Then, cross validation is implemented for each hand independently. To evaluate the potential of hand posture detection models trained with data from multiple sets of hands, the cross validation process is also performed on a dataset containing recordings from sessions two and three. The results for each session are shown in Table 1.

Session	HOG	HONV
P1 Realsense	94.8%	93.4%
P1 Kinect	92.4%	93.6%
P2 Kinect	97.2%	98.9%
Combined Kinect	93.7%	96.0%

Table 1. 5-fold cross validation accuracy averages for each session

To reduce the potential of overfitting our models and to meet the needs of the system as previously discussed, we next evaluate the performance of models trained with static hand postures. For this evaluation, each model is trained using only recordings of the static hand exercises **A** and **B**, about ten seconds of recordings in total. Predictions are made for each frame of all remaining exercises using the trained models. Posture detection accuracy rates of each exercise averaged over both hands using the HOG and HONV feature sets are shown in in Figures 3a and 3b. Figure 3c shows the results of detecting hand posture for each pianist, P1 and P2, using models trained with data combined from the static hand postures of both pianists recorded by the Kinect.

The results in 2.2 show some variation in accuracy between each session and exercise. This is due in part to a limitation of our data collection process. To guarantee data for each hand posture, the pianists were asked to deliberately play with specific hand postures for each exercise. This presented a challenge to the pianists since poor posture is not natural. To overcome this limitation, a larger user study is being developed with piano student. The students will perform naturally while a piano teacher annotates their performance over time.

4. CONCLUSIONS

This paper presents research towards a depth camera based hand posture detection system for virtual piano tutoring. The results of initial experiments show the effectiveness of posture detection models trained on individual users to detect different hand posture mistakes made by piano students. With further research we plan to explore the best methods for providing the feedback to students.

To account for shape and size variations in hands and pianos, we implement and test individualized detection models. This configuration requires a short recording of the piano space to initialize background subtraction. A short detection model training session is also required as part of the system setup. As demonstrated in our research, this process is not overly invasive. Training a successful detection model requires as little as ten seconds of static hand recordings for each posture category.

5. REFERENCES

[1] G. Percival, Y. Wang, and G. Tzanetakis, "Effective Use of Multimedia for Computer-assisted Musical Instrument Tutoring," in *Proc. of the Int. Workshop on Educational Multimedia and Multimedia Education*. NY, USA: ACM, 2007, pp. 67–76.

[2] S. Ferguson, "Learning Musical Instrument Skills Through Interactive Sonification," in *Proc. of the 2006 Conference on New Interfaces for Musical Expression*. Paris, France: IRCAM; Centre Pompidou, 2006, pp. 384–389.

[3] E. Schoonderwaldt, A. Askenfelt, and K. F. Hansen, "Design and implementation of automatic evaluation of recorder performance in IMUTUS," in *Proc. of the Int. Computer Music Conference*, 2005, pp. 97–103.

[4] K. Riley, E. E. Coons, and D. Marcarian, "The use of multimodal feedback in retraining complex technical skills of piano performance," *Medical Problems of Performing Artists*, vol. 20, no. 2, pp. 82–88, 2005.

[5] M. Li, P. Savvidou, B. Willis, and M. Skubic, "Using the Kinect to detect potentially harmful hand postures in pianists," in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual Int. Conference of the IEEE*, Aug 2014, pp. 762–765.

[6] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 169:1–169:10, Sep. 2014.

[7] M. Tits, J. Tilmanne, N. d'Alessandro, and M. M. Wanderley, "Feature Extraction and Expertise Analysis of Pianists' Motion-Captured Finger Gestures," in *Proc. of the 2015 Int. Computer Music Conference*, Denton, 2015, pp. 102–105.

[8] A. Hadjakos, F. Lefebvre-Albaret, and I. Toulouse, "Three methods for pianist hand assignment," in *6th Sound and Music Computing Conference*, 2009, pp. 321–326.

[9] A. Oka and M. Hashimoto, "Marker-less Piano Fingering Recognition Using Sequential Depth Images," in *Frontiers of Computer Vision, (FCV), 2013 19th Korea-Japan Joint Workshop on*, Jan 2013, pp. 1–4.

[10] A. Hadjakos, "Pianist motion capture with the kinect depth camera," in *Proc. of the Int. Conference on Sound and Music Computing, Copenhagen, Denmark*, 2012.

[11] H. Liang, J. Wang, Q. Sun, Y.-J. Liu, J. Yuan, J. Luo, and Y. He, "Barehanded Music: Real-time Hand Interaction for Virtual Piano," in *Proc. of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. NY, USA: ACM, 2016, pp. 87–94.

[12] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773 – 780, 2006.

[13] J. Canny, "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, June 2005, pp. 886–893 vol. 1.

[15] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision*. Berlin, Heidelberg: Springer, 2012, pp. 525–538.

A Fluid Chord Voicing Generator

Daniel Scanteianu
Stony Brook University
daniel.scanteianu@stonybrook.edu

Errick Jackson
Harvey Mudd College
ejackson@g.hmc.edu

Robert M. Keller
Harvey Mudd College
keller@cs.hmc.edu

ABSTRACT

We present an interface and methodology for voicing chords for jazz piano accompaniment. Rather than selecting from voicings that are drawn from a preset vocabulary or, if necessary, generated by a fixed algorithm, our system supports a broad set of parametric specifications. A random element permits variety in the generated voicing sequences, as well as providing for aural experimentation. Generated voicings can be saved to augment a vocabulary if desired. The probabilistic method selects notes based on a series of weightings and multipliers. The general framework is designed to accommodate voicings that mimic human pianist hands. This method of voicing generation thus provides a viable real time alternative to jazz accompaniment for the Impro-Visor program, expanding the range of available sounds.

1. INTRODUCTION

During solos in a typical jazz combo setting, all members will likely be improvising their part to a certain extent. Although chords are specified in almost all jazz pieces, it is up to the musician playing them to choose the chord voicings and tempos. While multiple approaches exist for comping or providing a soloist with chordal accompaniment, most of these approaches rely on existing piano voicings such as drop voicings and quartal voicings. Despite the availability of piano voicing generators that use predefined voicings, there has been limited development of systems to generate new voicings that can be customized based on a comprehensive set of user preferences.

2. RELATED WORK

There have been multiple approaches to generating a sequence of jazz piano voicings from a set of chord changes. Most of these approaches use predefined voicings, and calculate an optimal voicing sequence. Impro-visor [1] (the software we extended with our method) also features a chordal accompaniment facility that chooses voicings from a preset library. The user can select from open, closed, quartal and "shout" voicings. There are predefined voicings for all common chord types in all four categories of voicing types, and the software chooses the closest voicing to the current one within the voicing type selected. If no voicing is specified, or there is none that fits the required range limitations, a voicing is synthesized. In both cases there is an attempt to choose notes that form acceptable voice leading.

The voicings considered in [2] included: four-way close voicings, their "drop" alternatives (which moves selected notes down an octave), backing voicings, and five-note spread voicings. In order to implement a voice leading algorithm, a voicing distance metric was used that calculated the sum of the distances between the corresponding notes of a voicing and prospective next voicings, preferring voicings with lower inter-note distances. This approach built on the system specified in [3], wherein chords are voiced based on user selection of one of the voicing types (close, drop, spread) and then checked for violation of specific rules, such as low interval limit. Reference [4] also proposed a voicing system that voices chords in a similar way, but accounting for the melody in choosing the voicings.

A probabilistic system was proposed in [5] that used a Bayesian network to determine whether or not to use extended harmony and which notes to omit when voicing chords. While this approach provides a more flexible model, the actual output is limited to voicings that usually comprise four notes, and are relatively closed, as they are intended for the lower manual of an organ.

An important set of voicing parameters for voicing generation was described in [6], which specified the position of the chord (similar to the classical inversion), register of the chord, and wideness (range from the lowest note to the highest note) of the chord. The system specified also took input from the user on how many notes to use in a voicing, and generated a voicing accordingly.

Perhaps the widest range of voicing options is provided by [7], an open source program designed to generate MIDI backing tracks, which has parameters for voicing range, note span, a limiter for the number of notes per chord, inversions, and a user-modifiable method to enter chord voicings for later use. It also incorporates voice leading elements, but they do not appear to be easily controllable by the user.

The voicing generators discussed above are usually very limited in terms of user-accessible options, and generate voicings based on fixed structural rules. While these voicings are usually acceptable, the variety is limited, and therefore they do not always provide the sound of a pianist's voicings. In this paper, a probabilistic system for generating chord voicings is specified, together with a

user interface that allows for more direct specification of jazz voicings based on a wide variety of parameters.

3. MOTIVATION

The aim of our Fluid Voicing Generator (FVG) is to address the lack of user customizability in the precursor voicing generator for Impro-Visor [1], while creating jazz chord voicings that a pianist could reasonably play. In order to generate authentic sounding voicings, FVG employs a set of limiting parameters that model a pianist's hands, but which may be overridden to generate voicings beyond the ability of human pianists. It was designed with the intention of providing an extension to the piano chord accompaniment abilities of Impro-Visor. FVG was required to have a broad range of possible chord voicings that the user would be able to customize and select in order to fit the style of jazz piano accompaniment relevant to the piece being played by the software.

4. PRECURSOR ALGORITHM

In order to generate a chord voicing, the precursor software starts with two lists of available notes provided by the Impro-Visor user's vocabulary file. An example of the relevant parts of an entry in the vocabulary file is shown below. A separate style file specifies the desired type of voicing and the limits on the range of notes. Although the chord specification is relative to C as a root, transposition to other roots is done automatically.

```
(chord
 (name CM69)
 (pronounce C major six nine)
 (spell c e g a d)
 (color b f#)
 (priority d e a g c)
 (voicings
  (left-hand-A (type closed)(notes e g a d+))
  (left-hand-B (type closed)(notes g d+ e+ a+))
  (quartal (type open)(notes e a d+ g+))
  (shout-A (type shout)(notes e g a d+)
           (extension d++ g++ d+++))
  (shout-B (type shout)(notes g d+ e+ a+)
           (extension d++ g++ d+++))))
```

The "spelling" (notes in the chord), arranged by in order of "priority" (high priority notes are more desirable in a jazz chord voicing, usually, the 3rd and 7th scale degrees are first), and the "color tones" (notes that are sonorous with the chord tones but that are not in the chord).

5. DESIGN OF FVG

5.1 Note Choices

The Fluid Voicing Generator also uses the Impro-Visor vocabulary file. It chooses notes from a weighted list of all the possible notes. In order to choose a chord voicing, a sequence of weight modifying constraints is applied to the weighted list, and the weights are modified each time a note is chosen. Notes are chosen at random based on the weightings, within the range limit and other constraints established by the user.

5.2 Hand Settings

In order to generate natural-sounding voicings, the Fluid Voicing Generator is generally set to play voicings that nominally could be played by a pianist. This means that FVG must simulate two hands, each of which can stretch a distinct number of semitones, based on user preference. The user may also require a minimum distance between notes, useful for generating voicings with open character. Each hand plays a number of notes, usually between zero and five for each chord. The number of notes to be played is randomly set for each hand, and can vary chord to chord, between the minimum and maximum allowed number of notes set by the user. Additionally, there are settings for the lowest and highest note that can be played by each hand, in order to allow the range of the piano voicings to be constrained and prevent the piano voicings from overlapping with both the bass part and the solo part being played by the user or generated by the software in automatic improvisation. As in real life, the hands may be allowed to overlap and cross, and can shift positions within the allowed range.

5.3 Hand Motion

Pianists often choose voicings that move consecutively down or up the keyboard. In order to replicate such motion, there is a setting that allows the hands to move generally downward, or upward, with a certain degree of randomness allowing the hands to move more or less (or even change direction) based on user preference. Once the hand positioning is established, the notes to be played are randomly selected from within each hand's range, based on the weightings assigned to the notes in the range.

5.4 Voice Leading

One of the major requirements for the Fluid Voicing Generator was to generate smooth voice leading between chords. Voice leading involves choosing a voicing for a new chord so that it shares notes with the previous voicing, or has notes that are very close to the notes in the previous voicing. In order to increase the degree of voice leading, FVG has independent user-adjustable multipliers that increase the weights of notes shared between the previous and current chord, notes one semitone away, and notes two semitones away, from the previous voicing.

5.5 Voicing Settings

Pianists often change the types of voicings they choose in order to vary their sound, and conform to a specific style of music. In order to allow the user to control the sound of the chord voicings, several parameters are presented. The user can select what maximum weighting to apply to the notes in the chord's spelling, as well as how much to decrease the weight of notes that are lower in the priority chain. In addition, there are independent left and right hand settings for color note weighting. There is a setting to reduce the probability of a note being played if it is already played in another octave (in order to help ensure that multiple pitch classes in a chord are voiced. There are also settings to reduce the probability of notes a half step apart or a whole step apart being played at the same time (in order to achieve lower chord density without eliminating small intervals entirely). These can be overridden by setting a minimum interval that dictates the minimum number of semitones between the closest notes in the chord.

5.6 Conventional Voicing Rules

The Fluid Voicing Generator was designed to be able to accommodate requirements for standard voicing including voicing all the notes in the chord at least once, performing rootless voicings, and inverting minor ninths to major sevenths. When engaged, these controls override user preferences when necessary, preventing hand spread and note limits from limiting chord notes, as well as inverting minor ninths to major sevenths, even if this involves placing notes a half or whole step apart. In order to do this, the options to voice all notes and generate rootless voicings are applied before constraints are. The minor ninth inversion option is applied at the very end of the signal chain.

5.7 Implementation

Figure 1 shows a screen shot of the control panel through which the user can specify settings. The Fluid Voicing Generator is implemented as a set of classes within the Impro-Visor program, which is open-source and cross-platform (Java-based). FVG uses an array of all the MIDI piano notes as its weighted list of notes. Figure 2 shows the control flow for determining voicings, with left-to-right priority. Notes that are neither in the chord nor in

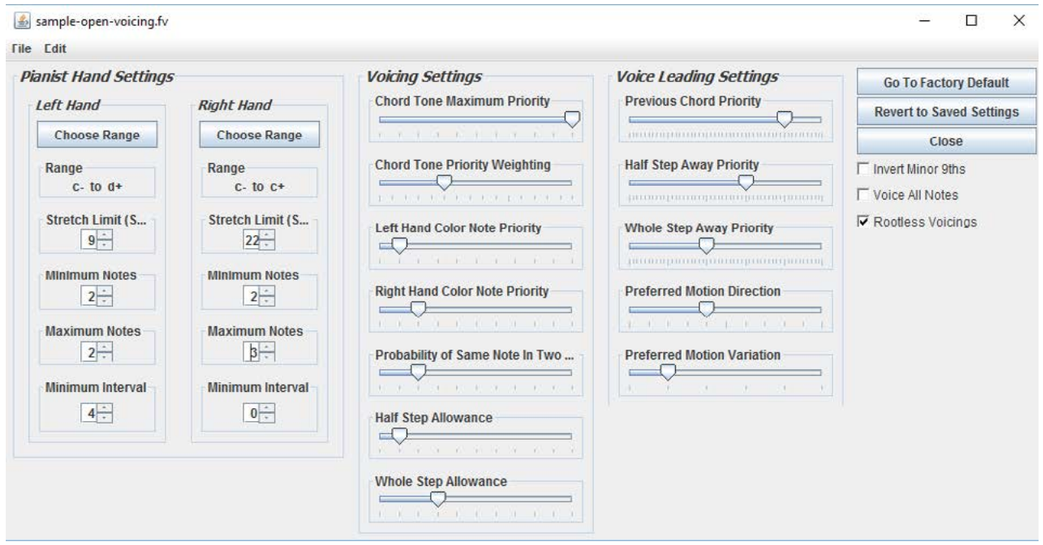
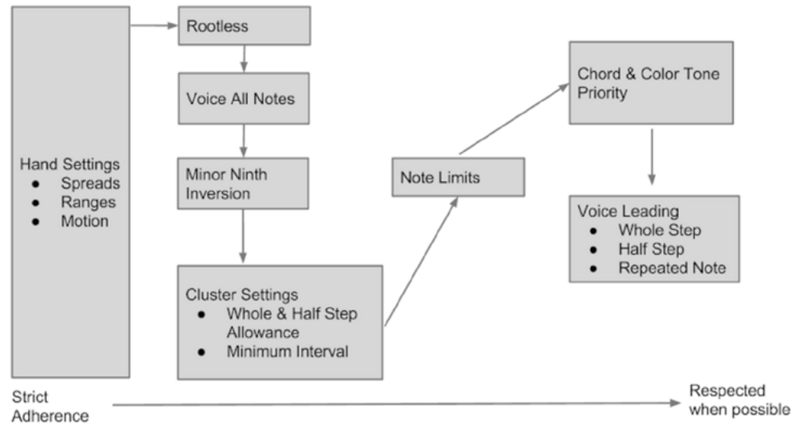


Figure 1. Control panel for user-specified voicing settings



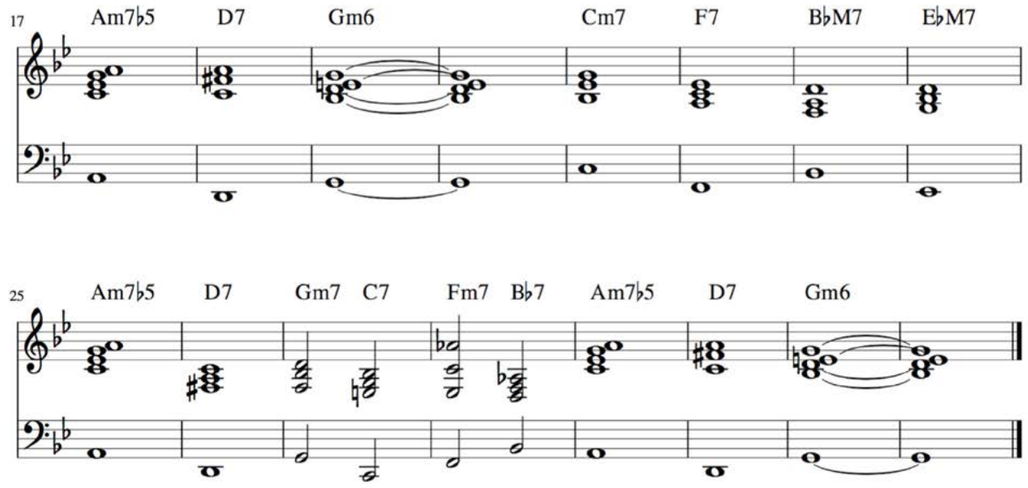


Figure 3. Example of the last 16 bars of "Autumn Leaves" with 3 note voicings using the precursor algorithm. Examples of voice leading discontinuities can be seen in bars 25-26 and 28.

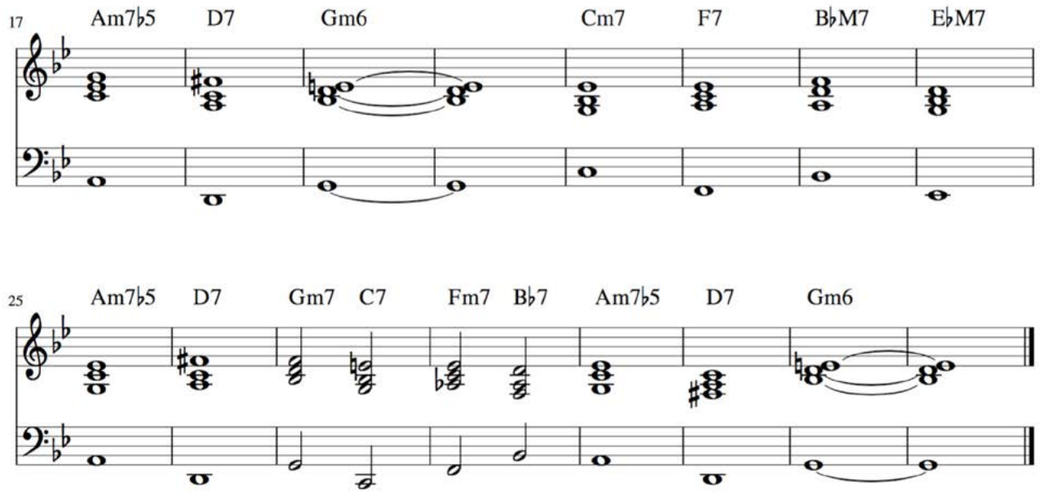


Figure 4. Example of the last 16 bars of "Autumn Leaves" chords with 3 note voicings using FVG. Voice leading is relatively free of discontinuities compared to Figure 3.

their extensions have their weights set to zero, and are thus unaffected by multipliers. Most weights are integer or have one decimal of precision, with all initial weightings ranging from 0 to 10 and multipliers ranging from 0 to 5. The implementation works in real time, generating new voicings for the entire chorus at the beginning of the chorus.

6. EVALUATION

A comparison between FVG and the precursor algorithm is demonstrated in Figures 3 and 4 using the last 16 bars of the chords for "Autumn Leaves". The precursor Impro-visor voicing generator generated voicings using its "closed" voicing setting (strongest voice leading available), and a range from the D below middle C to the A above Middle C. For the FVG version, chord notes are set to be based on previous voicings, with the priority weighting slider set low in order to generate smoother voice leading and directional motion. In qualitatively comparing the two figures, we see that the FVG rendering has no voicing discontinuities, whereas there are at least two in the precursor rendering.

For a quantitative comparison, we recorded each adjacent pair of voicings, the number of notes in each voicing, number of notes different in the voicings and sum of the difference between each note in one voicing and the closest note in the next voicing. These measurements were then summed and divided by the average number of notes between the two chords. The voicings varied between three and four notes with an average of 3.3 notes. The process was then repeated with FVG using similar range settings, flat directional settings, and three notes per chord. The FVG settings were set so that the voicings generated would be very close to traditional closed voicings (an octave of range, no repeated pitch classes, rootless voicings, high likelihood of high priority notes such as thirds, sevenths, and extensions).

With the Fluid Voicing Generator, the average probability that a note would change between two voicings was 0.67, and the average number of semitones of change was 1.67 semitones per note between two voicings. With the precursor algorithm's open voicing setting, the average probability that a note would change between two voicings was 0.75, and the average number of semitones changed between notes was 2.03 semitones per note between two voicings. A paired, one-tailed t-test showed that there was significantly less change in both categories at the alpha = 0.05 level for FVG with p = 0.016 for the probability of a note changing, and p = 0.001 for the average distance between a note and the nearest note in the next chord. This indicates that FVG is capable of stronger voice leading than an algorithm that tries to perform voice leading but requires traditional chord voicings.

7. FUTURE RESEARCH

To evaluate further how natural the Fluid Voicing Generator's voicings sound, settings that approximate conventional jazz voicings will be developed, with jazz experts invited to compare the voicings generated by FVG to conventional voicings and voicings played by famous jazz musicians. Furthermore, FVG settings should be expanded to allow the user to specify weights for the chord note subsets (root and fifth, third and seventh, etc.) individually for each hand, in order to create voicings with a specific structure and sound.

8. CONCLUSION

We have presented a method for arranging a jazz piano accompaniment by using a probabilistic jazz chord voicing generator. In tests using pieces in the standard jazz repertoire, the new voicings showed closer voice leading on average than preexisting voicings. However, unlike voicing generators that rely on preexisting voicings, multiple parameters can be extensively modified. The Fluid Voicing Generator can generate voicings that follow certain conventions, including voicing all notes, rootless voicings, inverting minor ninths to major sevenths, and keeping a minimum distance between notes. Additionally, the FVG can be set to generate voicings that would not generally be encountered in jazz. Such voicings can include more notes than a pianist can comfortably play, have further note spread than a pianist could manage, and voice the same pitch class in multiple octaves. The FVG provides voicings in real time, making it well suited to live accompaniment track generation. FVG has applications not only in computer generated jazz accompaniment, but also in assisting arrangements for piano, especially when an unconventional sound is required that follows certain voice leading, motion, and density criteria.

Acknowledgments

The authors thank Harvey Mudd College for its ongoing sponsorship of the Impro-Visor project. We thank Stephen Jones, who developed the precursor voicing algorithm for Impro-Visor, with which the Fluid Voicing Generator has been compared. This research was funded in part by the National Science Foundation CISE REU award # 1359170.

9. REFERENCES

[1] Robert Keller, Impro-Visor (Jazz Improvisation Advisor) https://www.cs.hmc.edu/~keller/jazz/improvisor/ (last consulted January 2016).

[2] Junko Watanabe, et al., "A system generating jazz-style chord sequences for solo piano," *Proceedings 10th International Conference on Music Perception and Cognition* (ICMPC), 2008, pp. 130-135.

[3] Norio Emura, et al., "A modular system yielding Jazz-style voicing for a given set of a melody and its chord name sequence," *Proceedings 19th International Congress on Acoustics,* Madrid, 2007, pp. 2134-2139.

[4] Norio Emura, et al., "Machine arrangement in modern jazz-style for a given melody," *Proceedings 9th International Conference on Music Perception and Cognition,* Bologna, 2006, pp. 110-115.

[5] Tetsuro Kitahara, et al., "Computational model for automatic chord voicing based on Bayesian network," *Proceedings 10th International Conference on Music Perception and Cognition* (ICMPC), 2008, pp. 395-398.

[6] Rui Dias, et al., "A computer-mediated interface for jazz piano comping," *Proceedings of the Joint 2014 International Computer Music and Sound and Music Computing Conference*, Athens, 2014, pp. 558-564.

[7] Bob VanDerPol, MMA—Musical MIDI Accompaniment, http://www.mellowood.ca/mma/ (last consulted January 2016.

How To Play the Piano

Richard Hoadley
Anglia Ruskin University
research@rhoadley.net

ABSTRACT

This paper describes work involving live generated audio and music notation and mapping techniques used between them. Live and prepared code as well as live and recorded sounds generate and manipulate the notated material which is then used synchronously in performance by a live musician. In this instance, the music notation generated is detailed common-practice notation, although in principal any arbitrary level of precision and expressive domain including text, graphics and images may be used. Text and audio of an original poem are manipulated algorithmically as a part of the composition. Decisions regarding what, when and how to play are taken by the performer before or during a performance, creating an environment which both interrogates the nature of notation and performance and provides a unique portrait of the the performer. Factors influencing all of these decisions are described and discussed as well as other works following these principles and the potential for future developments. This paper is presented alongside a submission of the live notation performance piece 'How To Play the Piano'.

1. INTRODUCTION

The work presented here is itself the result of a number of years' research, experiment and practice into algorithmic composition and the role of live notation within algorithmic environments. The impulse to explore these areas is the result of a series of creative factors, most importantly musical composition in which live performance as a significant element, as is the investigation of technology - and particularly computer technology - in music.

As all music performance is to an extent technological: involving a person manipulating an object for aural aesthetic ends - all music involves the mapping of physical actions onto audible results. Recent computer technology has enabled experiments into algorithmic mapping between domains such as audio, notation and text. The author has experimented with these in a number of recent compositions, performances and papers, for instance *Calder's Violin* [1] and the dance-text-music pieces *Quantum Canticorum* [2] and *Semaphore* [3]

Interaction between such creative domains has always existed in musical composition and performance [4]. Poetry is often accompanied by or has been an inspiration for music - an common mapping in musical practice for as long as music has existed - indeed, music may be an early form of verbal vocalisation [5]. Similarly, the use of visuals alongside music where one acts as the inspiration for the other are commonplace.

The popularity of such cross-domain investigations is to be expected as the technical means for creating such mappings has grown. Related mappings or translations include extra-musical elements not just onto audio via the musical imagination, or through the use of particular computer algorithms, but into notations which can then be performed and interpreted live and synchronously with other algorithmically generated material such as audio.

2. DYNAMIC NOTATIONS

The advent of live, dynamic notations is highlighted by the increasingly influential, if intangible, ramifications of the 'post-print' world [6]. This has had a particular effect on text both in terms of capability, reach and display, (for instance, Twitter, Facebook and Google Glass as respective examples), but technologies with similar capabilities in other expressive domains are becoming increasingly available and powerful.

How To Play the Piano utilises two pieces of software, the SuperCollider audio programming environment [1] and INSCORE , an environment for the design of interactive, augmented music scores [2] . INSCORE [7, 8] provides a particularly rich environment for a wide variety of dynamic notations. In addition to supporting the open source Guido music format [9], it supports MusicXML [10]. It also supports plain text, HTML and a variety of graphics formats such as SVG and PNG). The results of each of these 'languages' can be rendered immediately and can be graphically manipulated through code. The environment can be scripted internally or controlled externally via the Open Sound Control protocol (OSC), (see figure 2). As a part of the Interlude project [3] , it also enables cross-domain experimentation with gestures and interactivity.

2.1 Dynamic Text

'Liveness' has different consequences in different domains. For those working with text the ability of Google Docs to update material synchronously for all users is a literal demonstration of editing as performance. Inevitably artists have used this platform, alongside Twitter, as a way of interrogating particular methods of creating, viewing and performing with text [11].

Book publishing tends to emphasise the finished product - the messy processes of writing and editing are obscured by the impeccable published item. There have been a number of projects making use of electronic and networked resources, including novel-writing as performance [12] and as real-time performance [13], writing as performance art [14] and even Instagram [15].

Text can also be created and manipulated generatively rather than collaboratively [16]. This is less prevalent in text-based media although methods such as *Oulipo* [17] are well known and understood.

2.2 Automatic Notation Generators

The origins of live notation can be found in Automatic Notation Generators (ANGs). Those ANGs capable of live generation of musical notation have been around for a number of years - one of the first being Wulfson's *LiveScore* [18]. All of these seek to make a balance between flexibility, programmability and readability by the performer. Because of the inherently visual nature of most forms of notation there is often also an emphasis on visual aspects (although not all - some forms of wearables are now being used in versions of haptic notations). Kim-Boyle's *Tunings* [19] is a striking example of the use of visual processes to inform and challenge notation. More recently Ryan Ross Smith has created an extensive series of studies [4] each of which investigates the relationships between animated graphics and interpretation. Michael Edwards' *Slippery Chicken* environment retains close ties to traditional forms of notation delivery [20]. The development of an increasing number and variety of technologies in this areas demonstrate a growing level of interest which is also reflected in a number of comprehensive surveys of this area, in particular the recent Agostini and Ghisi [21] and Bean [22]. Further emphasising this growth in interest is Volume 29 of Contemporary Music Review (2010) which is entirely devoted to virtual scores and real-time playing as well as, of course, the inauguration of the annual TENOR (Technologies for Music Notation and Representation) conference in 2015.

The use of live ANGs draws attention to a number of compositional and performance related issues, such as the difference between improvising and playing from notation - including sight-reading, the importance of the presentation of material and the performer's relationship to it, the effectiveness of different kinds of notation, traditional or otherwise, and the best ways of dealing with instrumental synchronisation.

The history of fixed scores and the various modern reactions against them - the development of alternative notations, as well as entire alternative systems [23], however clear it becomes that they are unlikely to succeed [24] - perhaps blinds us to the obvious fact that "print is itself a medium [which may] be obscured by its long dominance with Western culture. As the era of print is passing, it is possible once again to see print in a comparative context with other textual media..." [25]

2.3 Live Notation and Algorithms

The conditioning of many hundreds of years of practice has left something of a vacuum between the fixed score and free improvisation. Notations generated algorithmically can find themselves anywhere on this continuum. While the Guido Library places less emphasis on the generation of the most complex scores involving deeply idiosyncratic and advanced notations it is designed, with delightful modesty, to be 'adequate' [9], in practice meaning that the necessary code is compact and easily transferrable via protocols such as OSC.

These systems can be highly responsive to other algorithmic streams, such as those involved in the generation of electronic music or the use of live sensed environments, such as in the music-dance-text piece *Semaphore* [3].

The use of such systems requires a system-wide balance between an arbitrary level of control of an arbitrary format. INSCORE is able to render both SVG files and streams, allowing algorithmic control over those resources as well.

Issues regarding the performers' interactions with the notations will be discussed in section 3.3.

3. PIANO GLYPHS: HOW TO PLAY THE PIANO

Piano Glyphs is an experimental piece for piano that is still in progress. It comprises several sections, of which at present only one - the main subject of this text - is complete. *Piano Glyphs* is an expressive investigation into the dynamic use of various types of notations: graphics, shapes and text as well as common practice music notation. The notations move, fade in and out, change opacity, colour, etc.

How To Play the Piano has been performed a number of times by two different pianists. One of the main points of interest to arise from the work is the way that different performers, and even the same performer in different circumstances or moods can interpret the music in a unique way.

3.1 Performances and interpretation

I originally met with Philip Mead to discuss the possibility of jointly presenting a lecture recital at the London International Piano Symposium 2015 based on my work with live notations [5] . We agreed that I would compose five *vignettes*, each displaying different musical characteristics of the technique. An example from one of these is shown in figure 1.

While somewhat sceptical of the notations which presented music in a more conventional way, Philip was intrigued by those which allowed him the musical space to improvise.

One of the questions with the implementation of live notation in *How To Play the Piano* is which live notation *mode* to use. The mode describes its intended use and presentation in performance.

Mode 1 is used primarily during composition itself. Notation is presented note by note, and as each note appears a synthesised rendering is played. In this way this mode also provides an audiovisual guide as to the intended *tempo* of

[1] SuperCollider is available here: http://supercollider.github.io

[2] INSCORE is available here: http://inscore.sourceforge.net

[3] http://interlude.ircam.fr/wordpress/

[4] many detailed here: http://ryanrosssmith.com/scores.html

[5] An audiovisual recording of a performance of the piece by Philip Mead is available here: https://vimeo.com/131433884

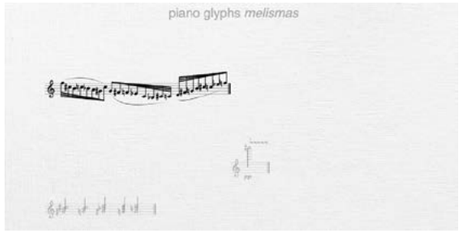


Figure 1. *Melismas* from *Piano Glyphs*.

the passage. In many cases, the notation represents what the algorithms suggest that the performer play and so **mode 2** is visually the same, but with no audio rendition. If the animation of the score's appearance is distracting, it is possible to present all notes in a particular phrase, or indeed an entire page at once, mimicking a traditional, physical page of music either with (**mode 3**) or without (**mode 4**) audio rendering.

Mode 5 is equivalent to **mode 3** but in which audio rendering is deliberately modified, allowing the performer to play the notation precisely as it is (for instance, without transposition). In the current system the notation appears the same whichever mode is being used, but it may be appropriate to identify the use of certain modes, for instance by the use of colour.

The use of modes provides a way of presenting material to the performer for rehearsal although the detail will be different for each rendering: a process which itself provides an novel perspective on live notation. I provided Philip with a number of easily shared fixed renderings (that is, screencasts of a 'performance' of the piece) during the process of developing and rehearsing the piece. The examples were first provided in mode 1, including a synthesised audio rendition of the notation. At first this troubled Philip: he assumed that the his role solely involved attempts to play the notation, in a sense in competition with the computer's rendition:

> "If [my function] is to simply try to play the music as it comes up there is no spontaneity as I know what's coming. The recorded sound does it in any case much better. I am instinctively drawn to using the dots for my own improvisation."
>
> Philip Mead *personal communication* [26]

After a number of performances it also became clear that Philip was under the impression that the score for each performance was rendered to a fixed audiovisual recording prior to each concert rather than being rendered live during performance. Subsequently we began a more interactive process of duetting in the moment - using live coding techniques to allow interaction between code, notation and performance.

Philip immediately made use of the text and its fragments as a way of inspiring improvisatory material. In a way that sometimes results in textures similar to those carefully crafted by Berio in the third movement of his *Sinfonia*, Philip would take a passage from Katharine Norman's poem [6] and use it as the basis for an *ad hoc* improvisation, usually, but not always, favouring the *Dance of the Swans* from Tchaikovsky's *Swan Lake*:

> *Miss Norman will play some skipping music,*
> *and fifteen infant ballerinas in pink and white*
> *will hop*
> *left leg, then right,*
> *across a cold church hall.*

Figure 2. A screenshot from *To Play the Piano*.

My session with Paul Jackson occurred later in the process by which time I understood that some live notation processes could be helped by more and earlier clarification in rehearsal. Paul came to the piece without much preparation or familiarity (although he had played an earlier live notation piece of mine the previous year).

With very little introductory discussion we played through the piece once, discussed the experience and played it again, recording both performances [7] . Differences in interpretation between performers were immediately apparent. Paul began by using the inside of the piano - a technique Philip has - perhaps unusually - never used in the piece. I felt this was a very appropriate technique and one that complemented the piece's audio component very well. In fact, Paul's performance in general felt a lot more as I originally *imagined* a performance of the piece to be like. This is in no way a criticism of Philip's interpretation. On the contrary, I have found working with Philip on the piece to be a particularly liberating experience.

3.2 The Poem

Katharine Norman's poem, *How To Play the Piano (in 88 Notes)* was written specially for this piece. As its title implies, it was written to reflect a concert piano and so the poem has 88 lines which can be divided into seven 'octaves' of 12 lines each and a 'minor third'. Katharine also provided an audio recording of herself reading the poem.

Apart from furnishing material to inspire improvisation, Katharine's poem also provides the major structural element of the piece. The first short section presents a growing mass of fragments of text, audio (both 'music' and 'speech') and music notation. The fragments of text are all taken from the poem and used purely as visual text, purely as audio (speech) or both. Similarly, the music is presented as visual and silent notation, as pure audio with no notation, or a notation with an audio version (using a synthesised piano sound). This audio version can be at pitch or transposed depending on the code used in performance as described above in section 3.1.

[6] The complete poem can be found at `http://www.novamara.com/how-to-play-the-piano`

[7] An audiovisual recording of Paul Jackson's performance is available here: `https://vimeo.com/132095458`

3.3 Performer Reaction

One of the most important aspects of this work is the reaction of performers themselves. Most classically trained performers are only too aware of the problems of new compositions - sometimes severe musical complexity or obscurity and often a desperate shortage of rehearsal time, of, as Lukas Foss commented on, "the precise notation which results in imprecise performance" and that "to learn to play the disorderly in orderly fashion is to multiply rehearsal time by one hundred" [27, p.45-53]. Performers are used to ensuring that however familiar they are with the complex music in front of them, they are able to give a genuinely impressive, expressive performance immediately. From this perspective live notation - at least live notation that is quite predictable in style and content - need be little different from standard fixed notation. This is, of course, an issue that composers need to implement carefully, particularly if elements of live coding are included. There is no real effort involved in algorithmically generating a phrase of notation, but a performer needs time to distinguish, ingest and react to material for it to be not just effective but also not irritating and confusing.

Neither are the more traditional alternatives always so attractive. Some performers have described frustration with the 'classic' electroacoustic performance paradigm of fixed score played alongside (rather than interactively integrated with) fixed sound track.

All performers who have experienced the system claim to enjoy it. Although some are apprehensive to begin with - unsurprising bearing in mind the novel experience and the improvisation/sight reading ostensibly involved - one performer mentioned that once they realised that even for myself as composer there were 'wrong notes' - they could relax more. The violinist Marcus Barcham-Stevens, who performed *Calder's Violin* in 2012, compared the experience to playing Brahms:

> "there was quite a large bandwidth of creativity and interpretation on my part - a different bandwidth to the degree of interpretation required when playing Brahms, but it is still uniquely strong, exciting and 'in the moment', given the degree of uncertainty (though within known parameters), and the need for total focus and concentration, possibly exceeding the concentration in 'standard performance', which inevitably gives the performance added tension and excitement." [8]

3.4 Projection and dissemination

The manner in which the generated music is presented to the performer(s) and whether and if so how it is conveyed to the audience is of particular importance [28]. In previous pieces such as those mentioned in section 1 above, the generated notation has been publicly displayed because general audience reaction seems to be in favour of this: a number of audience questionnaire responses to *Semaphore* [3] and related workshops make mention of the displayed notation. In these cases the instrumentalists involved played from laptop screens (although in some cases they also played at times, when convenient, from the public projection.

[8] Personal communication, 2012

4. FUTURE WORK AND CONCLUSIONS

While tautological, the most significant personal indicator of the validity and expressivity of these techniques and systems is that the author feels little or no impulse to use pre-composed music: if properly configured they allow for comprehensive flexibility with regard to the fixedness or otherwise of the material. Of course this is due to personal interests, but it is also because live notation provides the means of extending a continuing and long-standing interest in algorithmic music into the realm of music notations. Composers of all times, styles and convictions have made use of these processes in some way or another (Bach's elaborate canons from *Das Musikalische Opfer* or Mozart's *Musikalisches Würfelspiel* are frequently used early examples), just as performers use the equivalent in physicality - the use of 'automated' muscle memory during improvisation.

A new composition, *Edge Violations* for the clarinetist Ian Mitchell, computers and projections uses all the techniques described here including data acquisition via the Microsoft Kinect v2 and involving three back-projected screens on which the live generated score will be displayed. The score will include images, SVG graphics, text and music notation, all generated and/or manipulated live. In all these examples, the manner in which the material can be accessed by the performers is very important. Dancers, in particular, are hindered if they have to constantly refer to screens, however ubiquitous. It is possible, though not at all ideal, that optical head-mounted displays will provide some sort of solution to this.

Another project in development is the dance-music-text piece *Choreograms* - a collaboration between the author, the choreographer Jane Turner and the writer and poet Phil Terry. This interrogates the possibility of new live dance notations.

How To Play the Piano uses a form of live coding including a limited amount of live interaction between coding environment and performer through these notations. This is a potentially interesting area for development.

One of the most problematic areas to be resolved, should this be necessary, is the control of the amount of generated material and/or the rate at which this material is displayed for the performer. It should be possible to arrange systems using which the performer themselves are able to control, or at least influence this via some physical process.

5. REFERENCES

[1] R. Hoadley, "Calder's violin: Real-time notation and performance through musically expressive algorithms," in *Proceedings of International Computer Mu-*

sic Conference, ICMA, Ed. ICMA, 2012, pp. 188–193.

[2] ——, "Dynamic Music Notation in Quantum Canticorum," in *Proceedings of the 50th Artificial Intelligence and Simulation of Behaviour Conference*, R. K. et al., Ed., Goldsmiths, University of London, 2014.

[3] ——, "Semaphore: cross-domain expressive mapping with live notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, M. Battier, Ed., TENOR. Paris, France: Institut de Recherche en Musicologie, IReMus, 2015, pp. 48–57.

[4] W. Burnson, *Introducing Belle, Bonne, Sage*. Ann Arbor, Michigan: MPublishing, University of Michigan Library, 2010.

[5] S. J. Mithen, *The Singing Neanderthals*. London: Weidenfeld and Nicolson, 2005.

[6] R. Raley, *Comparative Textual Media: Transforming the Humanities in the Postprint Era*. Minneapolis, London: University of Minnesota Press, 2013, ch. 1. TXTual Practice.

[7] D. Fober, Y. Orlarey, and S. Letz, "Representation of musical computer processes," in *Proceedings of the ICMC/SMC 2014*, Athens, Greece, 2014.

[8] ——, "Augmented Interactive Scores for Music Creation." in *Proceedings of Korean Electro-Acoustic Music Society's 2014 Annual Conference [KEAMSAC2014]*, Seoul, Korea, 2014.

[9] H. Hoos, K. Hamel, K. Renz, and J. Killian, "The GUIDO Music Notation Format," in *Proceedings of the International Computer Music Conference*, ICMA, Ed., vol. 1998, ICMA. ICMA, 1998, pp. 451–454.

[10] R. Kainhofer, "A MusicXML Test Suite and a Discussion of Issues in MusicXML 2.0," in *Proceedings of the Linux Audio Conference*, Utrecht, Netherlands, May 2010.

[11] Performance Art Institute. (2012, October) The Artist is Elsewhere. [Online]. Available: http://theperformanceartinstitute.org/2012/07/index.html

[12] C. Ng. (2012) Novel-writing as performance art - http://fictionwritersreview.com/shoptalk/novel-writing-as-performance-art/. [Online]. Available: http://fictionwritersreview.com/shoptalk/novel-writing-as-performance-art/

[13] R. Sloan. (2009) Writing as real-time performance - http://snarkmarket.com/2009/3605. [Online]. Available: http://snarkmarket.com/2009/3605

[14] E. James. (2009) Writing As Performance Art - http://www.novelr.com/2009/10/10/writing-as-performance-art. [Online]. Available: http://www.novelr.com/2009/10/10/writing-as-performance-art

[15] A. Ullman, "Excellences and Perfections," 2014. [Online]. Available: http://webenact.rhizome.org/excellences-and-perfections

[16] J. Cayley, "Epigraphic Clock," website, January 2016. [Online]. Available: http://programmatology.shadoof.net/?clocks

[17] P. Terry, *Oulipoems 2*. Ontario, Canada: aha-dada books, 2009.

[18] H. Wulfson, G. Barrett, and M. Winter, "Automatic Notation Generators," in *Proceedings of New Interfaces for Musical Expression*, New York, 2007.

[19] D. Kim-Boyle, "Real-time Score Generation for Extensible Open Forms," *Contemporary Music Review*, vol. 29, no. 1, pp. 3–15, 2010.

[20] M. Edwards, "An Introduction to Slippery Chicken," in *Proceedings of International Computer Music Conference*, Ljubljana, 2012, pp. 349–356.

[21] A. Agostini, E. Daubresse, and D. Ghisi, "Cage: a high-level library for real-time computer-aided composition," in *Proceedings of the International Computer Music Conference*, ICMA, Ed., ICMA. Athens, Greece: ICMA, September 2014, pp. 308–313.

[22] J. Bean, "DENM (Dynamic Environmental Notation for Music): Introducing a Performance-Centric Music Notation Interface," in *Proceedings of the Technologies for Music Notation and Representation (TENOR) conference*, 2015.

[23] G. Read, *Source book of proposed music notation reforms*. Westport, Conn; London: Westport, Conn. ; London : Greenwood, 1987.

[24] R. Parncutt and G. McPherson, Eds., *Science and Psychology of Music Performance : Creative Strategies for Teaching and Learning*. Oxford ; New York: Oxford University Press, 2002.

[25] N. K. Hayles and J. Pressman, *Comparative Textual Media: Transforming the Humanities in the Postprint Era*. Minneapolis, London: University of Minnesota Press, 2013, ch. Introduction: Making, Critique: A Media Framework.

[26] P. Mead, "Personal Communication," January 2015, -.

[27] L. Foss, "The Changing Composer-Performer Relationship: A Monologue and a Dialogue," *Perspectives of New Music*, vol. 1, no. 2, pp. 45–53, Spring 1963.

[28] K. Norman, "Listening Together, Making Place," *Organised Sound*, vol. 17, no. 3, pp. 257–265, 2012.

Markov Networks for Free Improvisers

Stefano Kalonaris
Sonic Arts Research Centre
Queen's University, Belfast
Skalonaris01@qub.ac.uk

ABSTRACT

This paper discusses the use of probabilistic graphical models (PGMs) for initiating dynamical human musical interactions, in the context of free improvisation. This study proposes the model of Markov Networks and it speculates how they may serve for forming dynamical sepsets amongst players, based on their reciprocal beliefs, expressed as Bayesian inference. The prior is an assigned, private, musical personality. The players communicate their affinity preferences over a computer network using a graphical user interface. The conclusion is that Markov Networks viewed as dynamical Bayesian games are employable in the context of free improvisation and distributed creativity, providing a useful (and conceptually dissimilar) alternative to other structures that have been employed in music improvisation, such as graphic scores and idiom-based improvised forms.

Keywords: Probabilistic Graphical Models, Markov Networks, Music Games, Free Improvisation.

1. INTRODUCTION

The purpose of this research is to apply models of dynamical structural organisation based on probabilistic graphical models (PGMs) and Bayesian inference to game-based approaches in free improvisation. Although models derived from statistical, economic and computational sciences have been employed successfully in the areas of composition [1], algorithmic composition [2], [3] and machine improvisation [4], [5], there seems to be a shortage of studies that have addressed real-time interactions inspired by such mathematical and computational models in the context of free improvisation. Moreover, even in such cases, the conceptual framework has been that of a free improviser playing along/with an intelligent artificial counterpart. This paper proposes a model in which all players are human, a model that retains the performers' agency in the musical output, and where the machine is used for interfacing tasks only, as to provide a communication network within which the players operate. I claim that there is little if no historical precedent in this direction as all examples of Markov Networks (MNs) applied to music have been and are to be found in the areas of artificial intelligence and machine learning, be it applied to automatised generation of musical material (often *in the style of*) [6], [7], [8], [9], modelling musical structure [10], [11], statistical methods for audio processing [12], [13] and music information retrieval [14], amongst others. In contrast to these applications, I propose an abstraction for and between human players which is realised in real-time and, ultimately, with an associated freely improvised output.

2. FREE BAYES

Free music improvisation entails a high degree of dynamical shuffling of roles, which allows the participants to shape interactions in real-time and to react to unforeseen circumstances with split-second decision-making wizardry. This requires both the ability to make sense of the information available to them at any given time as well as the capacity to store and edit such information and beliefs in order to respond to their best. Such responses are based on the evaluation and the inferential analysis of the contextual evidence players are presented with. Such evidence is not immutable and static but, on the contrary, malleable and dynamic. Put simply, any improviser at any given point in time is actuating musical strategies that result from what she believes is happening or is going to happen in the near future. As soon as the player is provided with new evidence, she will adjust her response accordingly. This is analogous to what, in probability theory, is defined by the Bayes rule.

$$P(cause|observation) = \frac{P(observation|cause)P(cause)}{P(observation)} \quad (1)$$

The above reads: the probability of a cause, given the observation of an event, equals to the probability of the event given the cause, times the probability of the cause, all divided by the probability of the event.

3. MARKOV NETWORKS

Markov Networks (MNs) are undirected and possibly cyclic graphs. In the context of the natural interactions between free improvisers, MNs are more appropriate than directed graphs (Bayesian Networks), as they allow influences and inferences to flow in both directions. A formal definition can be stated as follows: a Markov Network is a random field S, which is a collection of indexed random variables (either discrete or continuous) where any variable $\boldsymbol{F_i}$ is independent of all other variables in S. Such a network also satisfies the Markov property, which states that no matter what path the system took to get to the current state, the transition probability from that state to the next will be independent from such path.

$$p_{(i,j)} = P(X = (n+1) = j \mid X(n) = i, \quad X(n\text{-}1), \ldots, X(0) \tag{2}$$

The simplest class of MNs is the pairwise MN, an example of which is depicted in the following example:

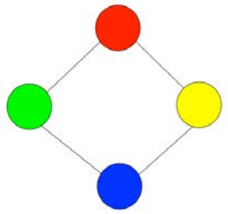


Figure 1. A pairwise Markov Network

In my musical implementation, the nodes of above graph represent four players, which, by virtue of playing together, influence each other. Since there is no strictly conditioning and/or conditioned variable, as one would have in a Bayesian Network, the notion of factor, hereinafter indicated as φ, will come in handy for defining the interactions between the nodes (players). Factors also go under the names of affinity functions, compatibility, soft constraints, and they generalise the idea of the local predisposition and willingness of any pair of nodes to take a joint assignment.

$\varphi_1[Red, Yellow]$

R^0	Y^0	30
R^0	Y^1	5
R^1	Y^0	1
R^1	Y^1	10

Table 1. Example of a local distribution amongst player Red and player Yellow

The above are arbitrary values and are chosen for illustration purposes only. The binary superscript (either zero or one) for the Red and Yellow players, in the respective columns, indicate their willingness to undertake a joint assignment with the other, or not. In the above example, the strongest factor indicates that neither Red nor Yellow would prefer to cooperate, talk to each other, play with each other, etc.
Similarly, one can imagine the other three local factors φ_2, φ_3, φ_4 and the probability distribution over the depicted pairwise MN would be:

$$\tilde{P}(Red, Green, Blue, Yellow) = \varphi_1(Red, Yellow)\varphi_2(Yellow, Blue)\varphi_3(Blue, Green)\varphi_4(Green, Red)$$

The above is not a proper probability distribution, since the sum over all the marginal distributions does not equal to one. In order to obtain a proper distribution, one needs to normalise by diving by the partition function Z.
The partition function is expressed as follows:

$$Z = \sum_{x \in \chi} \exp\left(\sum_k w_k^T f_k(x_{\{k\}})\right) \tag{3}$$

After having obtained the probability distribution, one can observe that the local preferences are no longer represented, as they have all been affected by the propagation of beliefs of all players over the network.
Put simply, even a four-player structure as this one, ends up in a complex aggregate of all the different factors that compose the MN. This is in contrast to what occurs in Bayesian Networks, where it is possible to inspect the probability distribution and retrieve a local factor. Pairwise MNs are not fully expressive and they are insufficient and inappropriate for representing all possible interactions. A more expressive model, used in my musical translation for improvisers, is the induced MN. In this model, each general factor ϕ has a scope that might contain more than two variables (as opposed to pairwise interactions).
A Gibbs Distribution is parameterised over a set of factors Φ, where

$$\Phi = \{\phi_1(D_1), \ldots, \phi_k(D_k)\} \tag{4}$$

The un-normalised probability distribution will be:

$$\tilde{P}_\Phi(X_1, \ldots, X_n) = \Pi_{i=1}^{k} \phi_i(D_i) \tag{5}$$

Whereas the normalised probability distribution will be expressed by:

$$P_\Phi(X_1, \ldots, X_n) = \frac{1}{Z_\Phi} \tilde{P}_\Phi(X_1, \ldots, X_n) \tag{6}$$

Where

$$Z_\Phi = \sum_{X_1, \ldots, X_n} \tilde{P}_\Phi(X_1, \ldots, X_n) \tag{7}$$

It is now possible to express a much wider range of scenarios, which might involve factors over three or more variables.

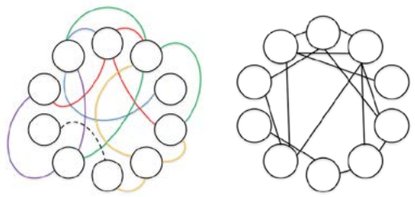


Figure 2. From local factors to induced Markov Network

Simply put, two or more variables (players, in this case) are connected whenever they appear in the same scope of a given factor. However, it would be impossible to infer the factorisation from the graph. In this sense, influence can flow along any active trail/edge. I find this model an exquisite abstraction of a typical interactive and dynamically assigned scenario amongst music improvisers, where alliances and joint assignments are formed, undertaken, updated, abandoned, in continuous real-time. Having understood the workings of an induced MN, I will now present my original rendition in musical form, as a dynamical model of interactions amongst free improvisers.

4. MN FOR FREE IMPROVISERS

4.1 Motivation

The decision of employing MN as a model for improvised musical interaction follows on previous experiments of mine, carried out in regards to focal points, Schelling's salience [15] and Markov Chains. The aforementioned experiments[1], pointed at the need to move towards an increased complexity of inter-relations and a decreased complexity of the instructions/constraints, as a step in the direction of allowing for more prompt and reactive environments for the player to operate in. Unlike the literature that has dealt with models based on either probabilistic graphical models or automata theory, Markov Network for Free Improvisers (hereinafter MN4FI) is an abstraction for and between human players and no musical output is generated by the machine. The hypothesis to be tested is to whether this given model provides alternative dynamical and interactive opportunities and modalities to groups of free improvisers, while maintaining freedom and flow in the performance.

4.2 System Design and Interaction Model

MN4FI follows directly from the example above regarding an induced MN. It formally maps players to a type and each type to a set of weighted strategies, or affinity preferences. The potential of this model lies in the fact that local distributions are not reflected or retrievable by the global graph and in that each of the players' screen might depict a different locality of connections.
The interaction model is described by a graph with up to ten vertices, each representing a different player. Each player is assigned a musical personality, what in Bayesian terms would be referred to as a *type*. Such type is private information and it is not shared amongst the players. This very fact implies that particular care needs to be taken when deciding on the spatial physical distribution of the players, in that they ought not to be able to see each other's screen.
Each of these four different types has an optimal local pairwise counter type, and ideally each player will try to infer the others' type in order to achieve such optimal joint assignment. Each player's degree (the number of other players he/she is connected to) is capped to *n-2* with *n* being the total number of players. Players can only musically interact with players they are connected to. The structuring principle consists in that each local graph might differ from any other, and the induced MN will not be common knowledge, nor will the local factor be retrievable should one be able to observe the resulting induced MN. Players are free to revoke one connection at a time, thus regaining the faculty of initiating a different one, if they wish. They instantiate and revoke edges according to their beliefs about the players they are connected or want to connect to. Additionally, players can also trigger a stochastic change of their own type.
The table below describes the affinity preferences for any local pair of players.

	Cooperative	Non-cooperative	Chaotic	Solipsistic
Cooperative	100	10	40	60
Non-cooperative	10	70	60	80
Chaotic	40	60	100	80
Solipsistic	60	80	80	100

Table 2. Inter-type local strategy matrix

It is important to note that the above numbers are arbitrary and the table is clearly not normalised.

4.3 Implementation and individual modules

MN4FI is realised in the programming language Max (`http://www.cycling74.com`). It follows a centralized design, consisting of one module for the players and one module which acts as a hub, receiving and despatching requests over a custom network, using the OSC protocol. There are *n* workstations, one for each player. At present, MN4FI accommodates up to ten vertices. In the presence of more than ten players, two or more of them can cluster around one workstation, thus sharing the screen, the assigned type, and the responsibility of connecting and removing edges.
In terms of actual coding, the core of the player's interface is realised using JavaScript within Max. This allows for a dynamical instantiation of the graph, depending on the number of players performing. Players connect and disconnect to and from a vertex by means of the numerical keyboard or using their GUI. They can also operate a trigger, which randomly reassigns their type. It is worth reminding that such type is private information.
The player patch has been compiled into a standalone application, in order to ensure that all players can run the interface, regardless of whether they have Max or not. Such consideration stemmed from the necessity to widen participation beyond limits of economical nature, Max being proprietary software. Each player's node will appear in red on their respective graph, and each node they are connected to will be coloured green. Else, the disconnected nodes appear in yellow. The GUI shown in fig.3 is what any given player sees and interacts with.

[1] partially available online at:
http://www.ransompaycheque.com/the-brazilian-games
http://www.ransompaycheque.com/finite-state-machines

Figure 3. The player's GUI

4.4 Evaluation

MN4FI was first played during the visit of Amsterdam-based duo Shackle at the Sonic Arts Research Centre (SARC) in Belfast on 03.12.2015, during which I had the opportunity to try the model out. At the time, MN4FI was implemented rather differently and it did not accommodate more than four players. MN4FI was subsequently reworked and tested with some of the members of QUBe, the resident experimental music ensemble of SARC. This time, MN4FI was played by eleven players. Both sessions have been recorded in audio and video format, and can be found online at the following website:

`http://ransompaycheque.com/markov-random-fields`

Fourteen players completed an evaluation form, as well as participating in short focus group discussions, over the course of the two instances. These were both valuable tools for obtaining feedback and suggestions for improvements, with respect to aesthetic, artistic and technological considerations. Given the small size of the sample (fifteen players in total), this paper can by no means claim to be conclusive or statistically significant.
The results obtained are, rather, a way to inform the next steps for the development of MN4FI.
The evaluation form is divided into three sections, each containing multiple questions to which the player can answer categorically on a Likert scale in 5 levels (from 'strongly disagree' to 'strongly agree', re-coded to 1-5). Players' proficiency was also reported in 5 categorical levels (from 'none' to 'expert'). The three sections present questions that address the degree of freedom experienced within the model, the degree of satisfactory output perceived, and how appropriate the design of the GUI is deemed, respectively. The answers collected pointed to the need of rehearsal time dedicated to familiarise and operate the GUI whilst maintaining the flow of musical improvisation. This is particularly true with respect to players who do not normally include electronics or other interfaces in their artistic practice. Proficiency levels were almost exclusively distributed between 'good' and 'expert' with 38.5% and 46.1% respectively. The remainder was evenly split between 'none' and 'proficient'. No player self-reported their proficiency as 'fair/basic'.

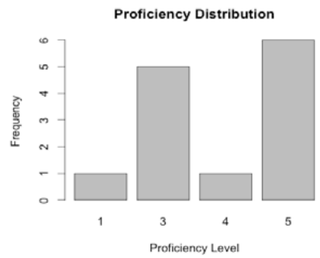


Figure 4. Proficiency levels

Levels of freedom experienced in playing this model were evenly distributed amongst level 3, 4 and 5, at 0.308, 0.385 and 0.308 respectively, leaving out the categories 1 and 2, which would correspond to a lower perceived freedom.

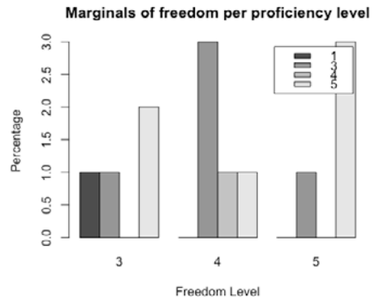


Figure 5. freedom with respect to proficiency levels

It appears clear that more experienced players had better chances to navigate the model with a higher likelihood of experiencing flow and un-hampered creativity in their performance. Overall, there was a consensus of the positive experience that all participants had of the piece.

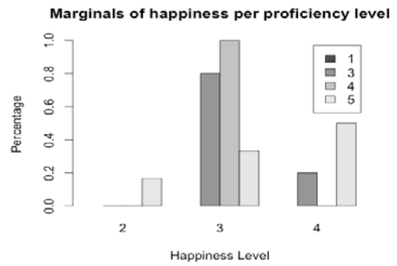


Figure 6. happiness with respect to proficiency levels

With regards to the evaluation in terms of inspiration for new ideas and interactions, the following are the percentages:

2	3	4	5
0.23076923	0.23076923	0.46153846	0.07692308

Table 3. Marginals for novel interaction

As seen from above, nearly 70% of the participants felt that the model suggested new and non-normative ideas (level 3 and 4). Furthermore, over 84% of the participants reported being happy and satisfied with the musical outcome.
From examining the correlation matrix for some variables in both the freedom and the output section, one can notice that, unsurprisingly, the strongest correlations are between freedom and constraint, proficiency and constraint, and between freedom experienced and the willingness to play again according to the model.

	Proficiency	Freedom	Novelty	Constraint	Play again
Proficiency	1.0	0.319	0.323	-0.412	-0.104
Freedom	0.319	1.0	-0.319	-0.795	0.532
Novelty	0.323	-0.319	1.0	0.022	0.234
Constraint	-0.412	-0.795	0.022	1.0	-0.334
Play again	-0.104	0.532	0.234	-0.334	1.0

Table 4. correlation matrix

The most valuable finding was, however, to be had during the focus group discussion, where it was reported that MN4FI encouraged a type of behaviour that was atypical, with respect to the simultaneous focus on both inner clusters of musical interaction and the global musical outcome.
" *I think it encourages a lot more interaction, like whenever we just play free improvisation people tend to go in their own wee world sometimes whereas with this, it kind of focuses you more on the fact that there are other people around you, also playing, and you have to listen to them".*
"Yes, it forms subgroups within something larger that is going on".
(QUBe members, focus group discussion, 23.02.2016)

5. CONCLUSIONS

In this work I have shown an equivalence between probabilistic graphical model based structures and Bayesian games in the context of a real-time interaction network amongst free improvisers. By implementing and testing a Markov Network as the determining structure for forming or abandoning musical local relationships amongst the performers, I have been able to show that insights from one area (PGMs) may be applied to the other (musical free improvisation) to provide an alternative and artistically valid and satisfactory modus operandi. I believe that this result is particularly exciting as it opens up numerous possibilities of intersection between free improvisation and paths that have so far been exclusive to the domains of decision theory, propositional logic and artificial intelligence. I claim to have employed a methodology that asserts the real-time human interaction as paramount, much in contrast to the uses of Markov Processes that have so far informed the discourse around musical improvisation and artificial intelligence/machine learning. In the latter cases, the Markov and Bayesian processes are employed to train an intelligent and autonomous artificial agent that either interacts with the human performer or generates music *in the style of*. Future work includes extending the model to allow players to send local assignments to their sepset and/or adapting the network to include more complex rules, for example in the form of Markov Logic Network, by the introduction of first order logic.

REFERENCES

[1] Xenakis, I., 1992. *Formalized Music: Thought and Mathematics in Composition*, Pendragon Press.

[2] Cope, D., 1991. *Computers and musical style*, A-R Editions.

[3] Cope, D., 2000. *The Algorithmic Composer*, A-R Editions.

[4] Lewis, G.E., 2000. Too Many Notes: Computers, Complexity and Culture in Voyager. *Leonardo Music Journal*, 10(2000), pp.33–39.

[5] Rowe, R., 2004. *Machine Musicianship*, MIT Press.

[6] Pachet, F., 2002. The Continuator: Musical Interaction With Style. *Proceedings of the International Computer Music Conference*, (September), pp.333–341.

[7] Pachet, F., Roy, P. & Barbieri, G., 2011. Finite-length Markov processes with constraints. *IJCAI International Joint Conference on Artificial Intelligence*, pp.635–642.

[8] Allan, M. & Williams, C.K.I., 2004. Harmonising Chorales by Probabilistic Inference. *Advances in Neural Information Processing Systems*.

[9] Assayag, G. & Dubnov, S., 2004. Using factor oracles for machine improvisation. *Soft Computing*, 8(9), pp.604–610.

[10] Mavromatis, P., 2005. The Echoi of Modern Greek Chant in written and Oral Transmission: A Computational Model and Its Cognitive Implications.

[11] Mavromatis, P., 2008. Minimum Description Length Modeling of Musical Structure. Journal of Mathematics and Music, 00(00), pp.1-21.

[12] Pearce, M.T. et al., 2010. Unsupervised statistical learning underpins computational, behavioural, and neural manifestations of musical expectation. NeuroImage, 50(1), pp.302-313.

[13] Pearce, M. & Wiggins, G, 2004. Improved Methods for Statistical Modelling of Monophonic Music. Journal of New Music Research, 33(4), pp.367-385.

[14] Cemgil, A.T. et al., 2008. Bayesian Statistical Methods for Audio and Music Processing, pp.1-45.

[15] Schelling, T., 1960. The Strategy of Conflict, Literary Licensing, LLC.

Opensemble:

A framework for collaborative algorithmic music

Matías Zabaljáuregui
Programa de Investigación: Sistemas Temporales y Síntesis Espacial de Sonido en el Arte Sonoro.
Escuela Universitaria de Artes. UNQ
matias@ventenmusic.com

Diego Dorado
Venten Music
diego@ventenmusic.com

ABSTRACT

This article introduces Opensemble, a framework for collaborative algorithmic music which employs the software engineering techniques used by the Open Source development model to discover the music that this kind of distributed and large scale collaboration can offer.

This proposal combines the author's prior research with current aesthetic concerns and an interest in exploring recent ideas behind Open Innovation for computer music. This work presents the first motivations of the framework.

1. INTRODUCTION

Opensemble is a framework created to explore the possibilities that arise when composing algorithmic music following the Open Source development principles. This paper is the first public document of our work in progress.

The pieces are entirely written using Supercollider programming language, with the use of GitHub as a collaborative platform and a model based on *trusted developers*, similar to that generated in the kernel Linux community [1]. Evaluation of the contributions is based solely on their technical qualities and artistic relevance towards the work at hand. A model of meritocracy is followed; a form of government in which hierarchical positions are reached based solely on merit [2].

The challenges to be met are many: how are the contributions organized? How is the work divided? What software engineering strategies can be incorporated into algorithmic composition? What effects does public communication during the development of the work produce?

Motivations for Opensemble stem from diverse sources of research, such as exploration into computer music, network music and applying the Open Source model as a paradigm for collaborative composition. Found below, are the concepts arising from these three disciplines which will allow us to define and outline our proposal with greater precision. For lack of space we will not deal with the designs based on tactile collaborative interfaces [3, 4], nor will we analyze laptop orchestras. [5, 6].

2. A COLLABORATIVE APPROACH FOR ALGORITHMIC COMPOSITION

There are many works which deal with the current and historical advances in algorithmic composition and computer assisted composition. Among the most distinct is that of [7], which focuses on Artificial Intelligence techniques and [8], which provides an introduction to the basic techniques and complementary examples of creative work. Collins [9] attempts to explore the possibilities for musical form in algorithmic composition from a psychological perspective and Nierhaus [10] performs a review of a great variety of methods used for algorithmic composition, with further investigation being done in [11] where a dialectical work performed by 12 composers and the different algorithmic composition techniques used is presented.

This work assumes that algorithmic composition will become an ever more complex activity. The implementation of hybrid systems which combine several algorithmic approaches has led to new possibilities of expression. Nevertheless, "*The main disadvantage of hybrid systems is that they are usually complicated, especially in the case of tightly-coupled or fully integrated models. The implementation, verification and validation is also time consuming."* [12]. An example of increasing complexity is shown by Musical Metacreation (MuMe) which combines several disciplines to study the automatization of musical creativity. In [13] the need is discussed: "*...to explore collaborative methodologies in order to make meaningful creative and technical contributions in the field. With the release of the Musebot specification, such opportunities are possible through an open-source, community based approach…*".

2.1 Motivations behind Opensemble

A small number of publications mention some methodological problems in related research. It is therefore interesting, in the initial stages of our project, for these to be reviewed so as to avoid reproducing the errors that are mentioned.

In Fernández's article [7] phrases such as "*reinventing the wheel in algorithmic composition techniques was common*" and *"...artists, who tended to develop ad hoc solutions, and the communication with computer scientists was difficult...*" stand out. Opensemble proposes practices, tools and standard protocols imported from Open Source which make communication easier and avoid the cost of having to create ad-hoc tools.

On the other hand, in Pearce's classic paper [14] , four possible high level motivations for writing software for musical composition are identified and the problems observed in literature as a result of being unable to distinguish the possible motivations, are later exposed. We have positioned our motivation among the first two mentioned by Pearce: in motivation 1, "*software written by the composer as an idiosyncratic extension to her own compositional processes"*, there are no methodological limitations and it is unnecessary to define rigorous criteria in order to define success, as the design of the software is part of the creative process. In motivation 2:"*software written as general tools to aid any composer in the composition of music"*, the problem becomes a software engineering task. Consequently, software engineering standards should be upheld.

Pearce encourages interdisciplinary work when there is more than a sole motivation, as is our case. Nevertheless, each motivation implies different methodologies and evaluation criteria. As a result, it will be necessary to take these overlapping motivations into consideration, in order to enrich interdisciplinary work and in turn define objectives and methods of evaluation for each.

An evident motivation in Opensemble relates to collaborative creation through distributed software engineering techniques. In this case "*Rather than exploring specific algorithms, this study focuses on system and component design."* [15]

Nevertheless, from the very beginning, our quest has been artistic/idiosyncratic. We consider that this work exposes a novel composition strategy, much as Open Innovation proposes new strategies of innovation in different fields [16, 17]. In the same way that algorithmic procedures for artistic creation were explored in the past, with no formal or theoretical objectives, we propose exploring modern software engineering techniques in order to discover the artistic results achievable. It is a *practical motivation*, as explained in [14].

3. OPEN SOURCE PARADIGM AS A FRAMEWORK FOR NETWORK MUSIC

There have been experiments performed with network music since the end of the 1970`s. Other artistic disciplines (*media art, net art and web art*) have indeed performed more thorough research in the past but it is our aim here to approach them from the perspective of algorithmic composition.

"*Designing and implementing a network music system supposes that new, meaningful sonic results can be achieved by collaboration over computer networks*" [18]. This is exactly the prime motivation behind our project. We can define Opensemble as a framework for implementing *interconnected musical networks*, as stated in [19]:" Making decisions about the motivations, social perspectives, and the network architectures are essential steps towards setting the framework for an effective musical network project."

Taking the exploratory work done in the field of network music into consideration, [18, 20, 21, 22], several "*popular conceptions of Internet music*", [23], can be observed. Here, special attention must be paid to '*Music that Uses the Internet to Enable Collaborative Composition or Performance*', as in the case of the FMOL Project [24].

Lancaster [25, 26] pose the question whether the current creation paradigm, more homogenous than the heterodox techniques used in the initial projects of this discipline, is an authentic musical need or a simple convenience. This transformation "*has brought with it aesthetic questions about the reason and evolution of this new genre.*" [26]. Although not oriented towards performance, Opensemble offers the maximum flexibility for collaborative composition allowing for synchronous as well as asynchronous approaches [18] and supporting both the sequential horizontal approaches, in which composers add consecutive fragments, as well as the vertical approach, which allows for overlapping of voices and sounds or modification of pre-existing material. [24]

3.1 Social Organization

It has been observed recently that "*Network music ensembles are uniquely positioned to deploy heterarchical technologies that enable them to address radical democratic concerns relating to communication structures and power distribution."* [27]. As it is later explained in this work, Opensemble is a clear example of this statement.

Two questions are central when classifying a project as Network Music:

- *What are the goals and motivations for designing a musical network?*
- *What are the social perspectives, architectures, and network topologies that can be used to address these goals and motivations?*

Because of its motivations, Opensemble is classified as a structure-based system (as opposed to process-based) "*In structure-based systems, the main goal of the interaction tends to focus on its outcome"* [19]. Composers and designers of such systems are usually more interested in aspects such as artistic vision and compositional arrangement instead of educational or social experience of members involved.

From the perspective of social organization, Opensemble mimics the kind of government used by the Linux development community: *"uses Peer Production methods and can be considered a Virtual Network Organization with a Peer Governance structure. [28]"*

Opensemble follows Open Source [29] institutional design and proposes *"a bazaar-like approach to coordination and leadership so as to allow a core team, trusted lieutenants, and other motivated contributors to emerge". "As a project matures, new key developers can emerge*

based on the concept of meritocracy [where] status is earned based on the merit of a developer's contributions" [2].
Due to dependence on the work of volunteers, recruiting and holding on to the collaborators is a crucial factor for the success of a project of its kind. Consequently, it is necessary to understand the factors that affect the motivation of the developers [30, 31]. "*Furthermore, the threat of forking limits the ability of project leaders to discipline members."* [32]. That is most likely the reason why a *"review of literature identifies governance as an area of significant interest in the open source research community."*[33].

4. FIRST PROTOTYPE. DESIGN AND IMPLEMENTATION

The first prototype, named "*LHCVMM, Large Hadron Collider Visual Music Machine*" [34], is a visual music project based on data generated by the Large Hadron Collider [35], the world's largest and most powerful particle accelerator, located at CERN in Switzerland. The goal is to translate the data generated by the ATLAS detector [36], one of the four major LHC experiments, into stimuli for the musical composition and performance.

Since we've begun, our intention has been to solve problems as they arise based on the needs of each individual project. Soon enough several questions emerged:

- *What common language could we adopt to describe a piece of music without tying it to a specific work?*
- *How to translate ATLAS data into a descriptive piece of music to drive both music and visuals?*
- *What will be the framework to manage collaborations in such a work?*

4.1 Adoption of a common language

Rather than creating a new language to describe a piece of music, we adopted Denis Smalley's Spectromorphology [37] defined as the perceived sonic footprint of a sound spectrum as it manifests in time.

As proposed by Manuella Blackburn in her paper [38], we use spectromorphology to create a sort of musical score consisting of sound unit events over time. In data-based musical pieces, the data driving the music needs to be translated to generate this score. Otherwise, the score is created during the composition stage.

We defined a spectromorphological vocabulary as well as the sound unit data structure. Each sound unit may have three phases respectively called onset, continuant and termination. Each phase is comprised of several properties describing its growth and motion, spectrum, and texture motion.

4.2 Data translation and Score composition

Publicly available ATLAS datasets, are queried with the help of pyROOT, a python extension module that allows us to sift through this data with ease. Collission events are then translated into sound units based on their kinematic properties and other characteristics of the overall process. This collection of sound units represents the aforementioned score which is streamed as OSC messages.

This strategy allow us to write the score on a spreadsheet, utilize a simple script to read each row as a sound unit, and stream them as OSC messages. Although initialy meant to prototype LCHVMM, multiple scores can be written with these tools, proving the design to be functional across musical pieces.

4.3 Framework and Collaborations

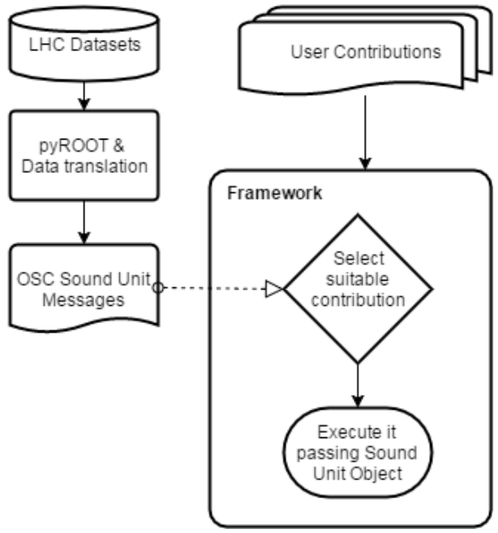
Having defined the vocabulary, the structure of sound units, the generation of scores and the mechanism to stream it through OSC, we must now exhibit how to turn this into playable collaborative music.

A framework was developed in Supercollider responsible for managing collaborations, receiving sound unit OSC messages, and finally reproducing the music. Collaborators register functions that can implement sonically sound units matching certain vocabulary terms by calling a method on the framework. Those functions receive a sound unit object as argument allowing collaborators to use its properties on their implementation. Finally, the framework, listening to OSC messages, selects the most suitable registered function for each received message and pass the sound unit object to the selected function. This selection is done based on the best match of vocabulary terms of function descriptions and sound unit object properties.

To recap, the framework performs the following actions:

1. All collaborations found in a special folder are registered.
2. An OSC listener is started to receive sound unit messages.
3. Upon message reception, a suitable function is selected.
4. The sound unit object is passed to this function to be performed.

Although we expect this design to change greatly as we progress, we are confident we are on the right track as it has already proven to be an efective approach.



5. CONCLUSIONS

There is no doubt that Opensemble presents a variety of compositional and technical challenges. The algorithmic composer will have to incorporate new practices and tools, will have to be open to discussing his ideas in public mailing lists and will have to accept that his work be published under open licenses.

We believe this to be a natural consequence and an inevitable convergence of the aforementioned areas of investigation. Moreover, it represents a motivation to reassess some of the loopholes in current computer music research. In this regard, a lack of articles in the area of software engineering applied to collaborative algorithmic composition stands out. It is particularly interesting to study the possible applications of *Distributed Agile Development* (DAD) which has received increasing interest both in industry and academia [39].
"*Characterized by a globally distributed developer force, a rapid, reliable software development process and a diversity of tools to support distributed collaborative development, effective FLOSS (Free/Libre and Open Source Software) development teams somehow profit from the advantages and overcome the challenges of distributed work"* [40]
One of the main challenges of our work is to discover ways in which to incorporate the advantages of Open Source to specific projects of collective musical creation based in the Internet. It is essential then to point out the need for an interdisciplinary perspective. Opensemble is an experiment that in turn encloses network music, algorithmic composition and Open Source software engineering.
The experiences with Opensemble in these first months have been encouraging. The collaboration has brought about interesting designs and we are persuaded that there is much interesting music to be found.

6. REFERENCES

[1] C. Schweik. "The institutional design of open source programming: Implications for addressing complex public policy and management problems", *First Monday* [Online], 8.1, 2003.

[2]. P. Ågerfalk, B. Fitzgerald, and K. Stol, "*Software Sourcing in the Age of Open: Leveraging the Unknown Workforce"*. Springer, 2015.

[3] N. Klügel, M. R. Frieß, G. Groh, and F. Echtler, "An approach to collaborative music composition", *Proceedings of the international conference on new interfaces for musical expression,* 2011, pp. 32-35.

[4] N. Klügel, A. Lindström, and G. Groh, "A genetic algorithm approach to collaborative music creation on a multi-touch table", *40th International Computer Music Conference, ICMC 2014, Joint with the 11th Sound and Music Computing Conference, SMC 2014,* 2014, pp. 286-292.

[5] D. Trueman, et al, "PLOrk: the Princeton laptop orchestra, year 1", *Proceedings of the international computer music conference,* 2006, pp. 443-450.

[6] G. Wang, et al, "Stanford laptop orchestra (slork)", *Ann Arbor, MI: Michigan Publishing, University of Michigan Library*, 2009.

[7] J. D. Fernández and F. Vico, "AI methods in algorithmic composition: A comprehensive survey", *Journal of Artificial Intelligence Research* (2013): 513-582.

[8] J. M. Peck, "Explorations in algorithmic composition: Systems of composition and examination of several original works", *Master's thesis, State University of New York, College at Oswego*, 2011.

[9] N. Collins, "Musical form and algorithmic composition", *Contemporary Music Review*, vol. 28, no. 1, pp. 103–114, 2009.

[10] G. Nierhaus, *Algorithmic Composition: Paradigms of Automated Music Generation*, Springer Berlin / Heidelberg, 2009.

[11] G. Nierhaus, *Patterns of Intuition - Musical Creativity in the Light of Algorithmic Composition*, ISBN: 978-94-017-9560-9 (Print) 978-94-017-9561-6 (Online), Springer, 2015.

[12] G.Papadopoulos and G. Wiggins, "AI methods for algorithmic composition: A survey, a critical view and future prospects", *In Proceedings of the Symposium on Musical Creativity*, pp. 110–117, 1999.

[13] A. Eigenfeldt, O. Bown, and B. Carey, "Collaborative Composition with Creative Systems: Reflections on the First Musebot Ensemble," *Proceedings of the Sixth*

International Conference on Computational Creativity, 2015

[14] M. Pearce, D.Meredith, and G. Wiggins, "Motivations and methodologies for automation of the compositional process", *Musicæ Scientiæ*, vol. 6, no. 2, pp. 119–147, 2002.

[15] C. Ariza, "An Open Design for Computer-Aided Algorithmic Music Composition: athenaCL", Ph.D. thesis, New York University, 2005.

[16] C. Herstatt and D. Ehls, *Open Source Innovation: The Phenomenon, Participant's Behaviour, Business Implications (Routledge Studies in Innovation, Organization and Technology)*, 2015.

[17] J. Faludi, "Open innovation in the performing arts. Examples from contemporary dance and theatre production", *Corvinus journal of sociology and social policy*, vol. 6, no. 1, pp. 47-70, 2015

[18] A. Barbosa, "Displaced soundscapes: A survey of network systems for music and sonic art creation." *Leonardo Music Journal,* vol. 13, pp. 53-59, (2003):

[19] G. Weinberg, "Interconnected musical networks: Toward a theoretical framework." *Computer Music Journal,* vol. 29, no. 2, 2005, pp. 23-39.

[20] M. Akkermann, "Computer Network Music Approximation to a far-scattered history", *Proceedings of the Electroacoustic Music Studies Network Conference - Electroacoustic Music Beyond Performance*, Berlin, 2014.

[21] M. Ayers, *Cybersounds: Essays on virtual music culture*, Peter Lang, 2006.

[22] W. Duckworth, *Virtual music: How the Web got wired for sound*. Routledge, 2005.

[23] A. Hugill, "Internet music: An introduction.", *Contemporary Music Review*, vol. 24, no. 6, 2005, pp. 429-437.

[24] S. Jordà, M. Alonso, and Grup de Tecnología Musical. "MÚSICA E INTERNET, CREACIÓN, INTERCAMBIO Y EDUCACIÓN", 2006.

[25] S. Gresham-Lancaster, "Computer Network Music", *Arts, humanities and complex networks*. Sci 2013. Retrieved from: http://ahcncompanion.info/abstract/computer-network-music.

[26] S. Gresham-Lancaster, "Computer Music Network", *Leonardo*, vol. 47, no.3, pp. 266-267, 2014.

[27] S. Knotts, "Changing Music's Constitution: Network Music and Radical Democratization", *Leonardo Music Journal*, vol. *25*, pp. 47-52, 2015.

[28] Linux Governance. Retrieved from: http://p2pfoundation.net/Linux_-_Governance

[29] E. S. Raymond, *The Cathedral & the Bazaar: Musings on linux and open source by an accidental revolutionary*, O'Reilly Media, Inc, 2001.

[30] D. Ehls, C. Herstatt, "Open Source Participation Behavior-A Review and Introduction of a Participation Lifecycle Model", *In 35th DRUID Celebration Conference*, 2013.

[31] G. Von Krogh, S. Haefliger, S. Spaeth, and M. W. Wallin, *Carrots and rainbows: Motivation and social practice in open source software development*, *MIS quarterly*, vol. *36, no.* 2, pp. 649-676, 2012.

[32] Y. Li, C. H. Tan, and H. H. Teo, "Leadership characteristics and developers' motivation in open source software development", *Information & Management*, vol. *49, no.* 5, pp. 257-267, 2012.

[33] A. Blekh, "Governance and organizational sponsorship as success factors in free/libre and open source software development: An empirical investigation using structural equation modeling. Doctoral dissertation". Nova Southeastern University. Retrieved from NSUWorks, Graduate School of Computer and Information Sciences. http://nsuworks.nova.edu/gscis_etd/40. 2015.

[34] Large Hadron Collider Visual Music Machine. Available at: https://github.com/Opensemble/lhcvmm/

[35] The Large Hadron Collider. Available at: http://home.cern/topics/large-hadron-collider

[36] About the ATLAS Experiment. Available at: http://atlas.cern/discover/about

[37] D. Smalley, "Spectro-morphology and Structuring Processes", *EMMERSON, S. The Language of Electroacoustic Music London, Macmillan Press Ltd*, 1986, pp. 61-93.

[38] M. Blackburn, "Composing from spectromorphological vocabulary: proposed application, pedagogy and metadata" , *Novars Research Centre, The University of Manchester.*
Retrieved from : http://www.ems-network.org/ems09/papers/blackburn.pdf

[39] M. Paasivaara, S. Durasiewicz and C. Lassenius. "Using scrum in distributed agile development: A multiple case study." *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on* 13 Jul. 2009: 195-204.

[40] K. Crowston, K. Wei, J. Howison and A. Wiggins, "Free/Libre open-source software development: What we know and what we do not know", *ACM Computing Surveys (CSUR)*, vol. *44, no.* 2, pp. 7, 2012.

Designing a Digital Gamelan

Adrien L' Honoré Naber
Designer DIGIGAM
taro.a3n@gmail.com

ABSTRACT

DIGIGAM is being developed as a digital instrument to explore the possibilities of a digitalized gamelan. This project emerged from a common interest in both traditional and modern art forms and the wish to develop a midi controller based on instruments in Javanese and Sundanese gamelan ensembles.

The goal is to design a controller based on Indonesian instruments with the "Bonang" as a starting point. This is the most versatile instrument in the ensemble and also the most challenging to design. Gunawan [1.3] uses nine different articulations to get a variation of sounds from this instrument. These articulations will have to be translated without affecting the style of playing this instrument too much. Although we want to simulate the original playing style, we would also like to add options that are relevant in digital music and controllers.

We compare available technology and search for translations. How can you make optimal use of the available technology and how can you integrate this in the playing style? The design of the controller is based on modules so different elements can be changed or added for future adaptations of this controller.

1. INTRODUCTION OF PARTNERS AND STARTING POINT

1.1 Introduction

Is the sky local? In this case they sky is not local. The Netherlands and Indonesia are far apart but share a long history together. This project is about crossing
more than geopolitical borders is about collaboration between different cultures and disciplines, and we are constantly working on translations. Trying to create understanding for something different is the key element in our project. We combine each other's knowledge to make a controller based on gamelan and love to learn from each other's differences in search for a center where all aspects are equally important. We may remain local but we learn more when we search for unknown things. We did research on different controllers and interfaces in order to find out what would be most usable for simulating gamelan.

Apart from a couple of string instruments and flutes, gamelan is based on melodic percussive instruments [1]. For that reason, electronic DIY drum kits were a good starting point for our research.

1.2 Partners

LeineRoebana is known from shows like 'Ghost Track, Snow in June, Smell of Bliss and others'. For the show 'Ghost Track' LeineRobana collaborated with Indonesian composer Iwan Gunawan. Gunawan is from Bandung, Indonesia and is the composer of 'Ghost Track' and conductor of Ensemble Kyai Fatahilla. Gunawan is always searching for ways of combining gamelan with other forms of music and technology. Alexander Dijk is a developer of tactile controllers and graduated in developing a first prototype of a gamelan controller. I am a composer / producer and have been going to Indonesia since 2008. I have spent one and a half year there travelling, and researching Indonesian music, language and culture. I have built up a new network of artists and collected a lot of field recordings that I use for my own compositions. Gunawan and me met in 2013 when I was doing an internship in Indonesia.

This was the starting point of our musical relation. I was searching for musicians and music to record and Gunawan was searching for someone who wanted to create a sound library of a gamelan set. Gunawan came to the Netherlands some months later to perform the show 'Ghost Track' with LeineRoebana. While playing 'Ghost Track', ideas for a~~n~~ digital gamelan came in mind and Harijono Roebana of LeineRoebana and Gunawan contacted the faculty of Music Technology in Hilversum. Gerard van Wolferen and Hans Timmermans had the opportunity to borrow the gamelan set from Jurien Slichter from Ensemble Gending. This was the point where Alexander Dijk joined the team and the start of the DIGIGAM project.

1.3 Iwan Gunawan

Gunawan is a composer from Bandung, West-Java, Indonesia. He grew up with Sundanese gamelan surrounding him. He is a music teacher at Universitas Pendidikan Indonesia in Bandung. This is also the homebase of his ensemble Kyia Fatahilla and the place where I have met him for the first time. Gunawan differs from other gamelan composers in Indonesia. He is always searching for new opportunities to involve gamelan into other modern art forms and digital cultures. His wish was to make a

sound library of a gamelan that could be used as a tool for composing his music. Though some libraries are already available from SonicCouture and Sample Logic, these are based on Balinese gamelan and western music and are to be interpreted in a different way.

1.4 My ambition for this project

Both my grandparents and parents were intrigued by Indonesia. Pictures and pieces of art were found all around the house I grew up in. My parents regularly had people visiting from across the globe, also from Indonesia. As a child this was my first contact with the country and initiated my interests in exotic adventures. In my teenage years I got into electronic music, especially Triphop, Jungle, Drum & Bass and Dub music. When I became older I was looking for ways to combine my passion for travel and music. I wanted to go to the High school of Arts Utrecht to study Music Technology, but I needed a good plan to get allowed in. I decided to travel to Indonesia for four months (what turned into a year) and record sounds, music, and learn from the culture. Upon arrival in Indonesia I immediately understood why my parents and grandparents loved this county. This was the point in my life where I had decided I want to dedicate my carreer to music. Ever Since I traveled through Indonesia, the country and its culture became a big part of my life, this shaped my goals, my music and myself until this day.

2. WHAT IS GAMELAN?

Gamelan are gong ensembles. They are found throughout China and South-East Asia but, the main focus of this project is based on Sundanese and Javanese ensembles. These Indonesian gong ensembles are called 'gamelan'. Gam-e-lan means 'to hammer' in Indonesian language [1] and comes from the hammer shaped mallets the gamelan is played with. The ensemble consists of gongs and kettles in various sizes forged with precious metals like bronze, copper and gold. These instruments are tuned in two main tuning systems 'slendro' and 'pelog', although different tuning systems do exist. Slendro is pentatonic, pelog is diatonic. The gamelan sets are made in various villages across Indonesia and vary slightly in tuning. This is the characteristic that a village can give to the gamelan set, their fingerprint. These villages usually tune their gamelan sets to one main gong called the 'Gong Ageng'. This knowledge of forging gamelan sets is being passed over from generation to generation. Therefore, a wide variation in gamelan remains.

2.1 Understanding gamelan

In the design of the physical controller we have to keep in mind certain aesthetics. Gamelan is music that was played in Courts and accompanies the Dance Theater [1]. The epic Hindu stories of Ramayana and Mahabharata are good examples. Music, dance and theater are usually intertwined with each other. You cannot think about one without the other. This synergy between the different disciplines is something that inspired us all individually and finally led us to initiate our research together.

3. REASON FOR A DIGITAL GAMELAN

3.1 Design criteria:

- Maintain the style that the instrument is played in, as much as possible;
- Better transportability. A gamelan set is heavy and difficult to transport and not always available. The DIGIGAM must be easy to transport;
- Plug & play, only connect to a computer with the right software.
- Intergrade digital effects and controls to extend the playing style in a way that the instrument will be useful for composition and improvisation;
- Tasteful design. Because this instrument will become part of a show we have to keep in mind the design of the show.

3.2 Extended sounds

Gunawan uses a lot of new techniques to extend the play style of gamelan instruments. He also combines the slendro and pelog system in his compositions. Gunawan makes use of every surface of the instruments and mallets. During our recording in the anechoic room we made a video documentation of all the different articulations being made. This information we use as a reference to organize the sample player and the controllers design.

Articulations on instruments:
- Long
- Short
- Short 2 (different damping)
- Very short
- Very short 2 (different damping)

Extended articulations:
- Using the side of the bonang's and gongs FX 1 & 2
- Using the end of the stick or mallet to hit and damp the bonang's and gongs FX 3
- Rubbing the bonang's with their fingers in order to get the pan singing FX 4
- And the musicians use their mouth as a modulator for the bonang and gender FX 5

4. DESIGNING THE SYSTEM

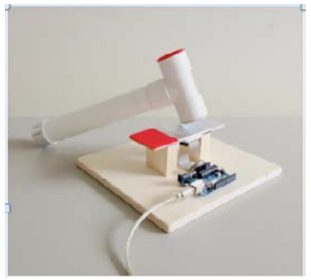
4.1 Recording

The recording of the gamelan set in 2013 was the first step in creation of a digital gamelan. We had the opportunity to record the gamelan set in an anechoic room. We have recorded with 3 microphones (AKG C-414, AKG C-451 and a DPA 4090) Recording in an anechoic room gave a lot of fine surprises and the sound quality of the samples turned out to be a treat to compose with. We have recorded 18 gong's, 2 gender's, 2 selentem's, 6 saron's, 4 bonang's and 16 kenong's, a total of 1261 samples in three velocities. Then this many samples needed to be very organized, so structure in the library is vital. I have categorized this in: tuning > instrument > instrument type.

4.2 Hardware:

During the recording process we made an observational research on the different methods of playing on the gamelan instruments. We registered the different instruments and articulations on video as a point of reference. There have been regular contact and testing sessions, trough trial and error we made different designs and tested different technologies. We have compared the playing methods used in gamelan with other digital instruments and available controllers like the 'Gamelan Elektrika'. Most similar to the gamelan were the electronic drum kits, especially the snare drums. These round triggers are a comparable size and can function like a gamelan, if modified.

Alexander Dijk wrote a supportive narrative about 'Digital Gamelan'. There he explored the possibilities of designing gamelan controllers. He did different experiments and ended up with a prototype for the gender, saron and selentem. **(Figure 1)** His study involved making a controller that mimics the behavior of these instruments. He researched the behavior of the original instrument and compared this to available technologies and sensors. The end result was a playable controller note where you have different options for triggering sounds. In a later stage this selected technology was implemented in a prototype design of the 'bonang' **(Figure 2).**

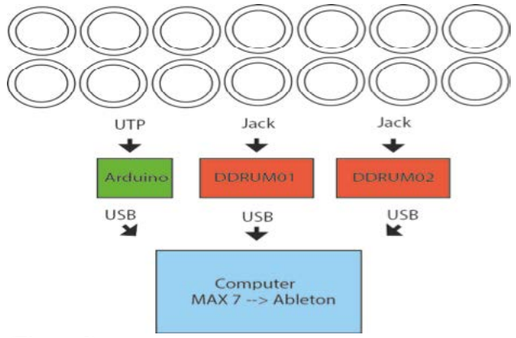


(Figure 1)

(Figure 2)

Making our own controller was an essential process in designing this instrument. However, it felt like we were 're-inventing the wheel'. These prototypes reacted like we wanted, but the system became more complex while our goal was efficiency. These realizations made us look back at available technologies and we started testing with a Yamaha electronic drum kit. These test worked out well for Gunawan and we decided to continue our research in modifying existing drum controllers.

The bonang uses a maximum of nine different articulations on a maximum of fourteen kettles. The challenge was here how to design a controller were you can play these samples within reach. For this we chose the Roland PD 8 dual trigger pad. The PD-8 includes two sensors a piezo and a switch. Each trigger pad is equipped with a FSR pressure sensor to register damping. We solved the length of the notes with a long sample being played, the FSR controls the sustain and release of the sample. When it is pressed, the sustain and release close, thus making the sound shorter. We equipped the controller with additional capacitive sensors in order to connect movement to audio effects so that the player can influence the color of the sound. The sensors are connected with UTP cables to the Arduino, these cables are easily connected, making it easy to build up the instrument.

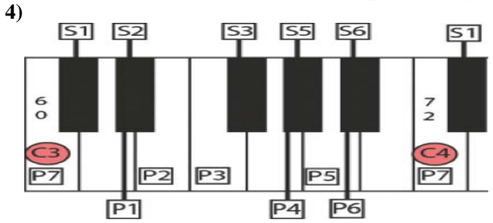


(Figure 3).

We based the instrument on modules that are connected to a 'brain' **(Figure 3).** For this, a couple of translations were to be made. The drum triggers have a stereo jack output and were routed through two DDRUM DDTI Trigger interfaces. We discovered that midi is fastest when the information is sent by USB. With the DDRUM DDTI the data of the drum triggers are converted inside and sent out by USB. With this we can keep the system efficient, saving CPU power. Arduino processes the data of the FSR and Proximity sensors; this data is then transmitted trough UTP cables. This connects to the computer system trough USB as well. With our philosophy of creating an instrument in modules, we created a system that is very adaptable and adjustable for different situations and can function for more than only a digital gamelan controller.

4.3 Software:

With many digital samplers available we decided to choose KONTAKT for composing, because of its wide options of modifying your digital instrument. Both Gunawan and me were familiar with this sampler and wanted to learn more about the software. The scripting editor allows you to create most things you can imagine. To translate the gamelan set into a MIDI keys, I programmed the notes closest to the actual notes being played. **(Figure 4)**



(Figure 4)

Slendro 5 is the same tune as pelog 4 and slendro 6 is the same tune as pelog 6. Gunawan uses these notes to switch between the different tuning systems.

For composing, you want the note you play on your keyboard to be the actual key. For this reason, we have chosen to use the actual notes and keep it structured with one tuning in one octave. KONTAKT has the option to use multiple instruments within one sampler and control them with different midi tracks, within the DAW you use. This enables the composer to combine sets of instruments in the different tuning systems to their liking.

For performing however we analyze the data coming from the DIGIGAM controller we have to sort out how to control the different articulations and instruments. For this we wrote a MAX patch, this patch has 2 times 7 switches connected to the rim switches. This enables the user to switch between different articulations and instruments. The upper row pads 1 to 7 control different instruments by activating the channels in Ableton. The lower row pads control the articulations by pressing the rim this enables different MIDI pitch shifters that act like keyswitches. Another patch translates the values of the FSR sensors to data that controls the “sustain and release” in Ableton.

In Ableton I have sorted the notes in separate samplers, because we need the FSR data to damp separate notes. This is an “instrument rack” for group modifications and effects, within this is another “instrument rack” with a maximum of fourteen samplers inside to control the “sustain and release” of the single notes. The pitch enveloppes and pitchbends are connected to Proximity sensor 2 so it can be controlled by waving your hand over the sensor and modifying the sound. Proximity sensor 2 is connected to sendbus A in Ableton to control an effect. An additional Korg Nano controller is added to control a looper function.

5. IS THE SKY LOCAL?

The strength of this project lays in our differences and getting understanding for each other’s discipline and culture. We puzzled and brainstormed for a period of two years and made a lot of adaptations to the instrument and music system. This is an ongoing process and to create this instrument we need to communicate a lot in order to get understanding for each other. This mentality is vital in this cooperation between different cultures. To expand our mind, I think we have to venture into the unknown. Here we can discover new possibilities that life has to offer outside of our comfort zone. I think we should create more understanding for other people and the things we do not know yet.

The beauty of living nowadays is the technology, with a laptop I can take my work everywhere. This allows me to work outside my home where I receive different impulses that inspire me. I tend to search for opinions that are different than mine. It fascinates me how much you can learn from people that do not share your world vision. This is the main reason I like to working with multicultural groups. Therefore, as an answer to the question ‘Is the sky local?’: No, not to me

6. CONCLUSIONS

Like Lego, this instrument has the option to grow in various shapes and sizes. We can implement different technologies and experiments in realms first unknown to the gamelan society. We combine our knowledge to develop an instrument that can be included in ensembles but also serves as a solo instrument. This vision however makes that almost everything will be possible. In this we have to cut back and search for solutions that are relative and functional to gamelan and digital controllers.

Although this is an ongoing project, we do have a clear view on what needs to happen and how we can achieve this as a team. We have collected a lot of information concerning technology, Indonesian music and culture. Music Technology is slowly developing in Indonesia, Alexander Dijk and me share our knowledge about technology with Gunawan and Kyai Fatahilla. We have had days when we designed systems together with Gunawan and tested the updates.

Gunawan and Kyai Fatahilla are the end users in this and we have to customize the design to their logic. A playable version of the DIGIGAM controller will be made for them in addition to an original gamelan set and will be finished by May 2016 for the LeineRoebana show “Light”. Gunawan and his ensemble need to understand how the instrument works. Therefore communication and involving them in creative processes remains essential.

Acknowledgments:
LeineRoebana, Iwan Gunawan, Kyai Fatahilla and Alexander Dijk formed an essential part in the realization, this project isn’t possible without them.

Reference:

[1] Henry Spiller -Focus; Gamelan Muisc of Indonesia (second edition, 2008)

Wavefolding: Modulation of Adjustable Symmetry in Sawtooth and Triangular Waveforms

Dr Edward Kelly
University of the Arts London
Camberwell College of Arts
Peckham Road
London SE5 8UF
United Kingdom
synchroma@gmail.com

ABSTRACT

The Pulse-Width Modulation (PWM) technique has been used to generate varying timbres of odd-harmonic spectra from early on in voltage controlled analog synthesis history. Methods for controlling the symmetry of a triangle-to-sawtooth wave have also been devised. This paper discusses a family of objects and techniques for piecewise waveform manipulation that may be modulated at audio rate, comparing the results with analog equivalents, and looking specifically at the implications of modulator phase and subtle deviations from integer carrier-to-modulator ratios, and fine deviations from these, on adjustable-symmetry sawtooth waves.

1. INTRODUCTION

The sawtooth or ramp wave is a fundamental element in subtractive synthesis, since it contains both odd and even harmonics of the fundamental frequency. It's slightly dull cousin, the triangle wave, has weak overtones of odd harmonics and sounds much like a digital approximation of a sine wave. Both have their uses in synthesis, but it is possible in both analog and digital domains to generate waveforms that can be modulated between sawtooth and triangle. Some digital synthesis methods have used this principle particularly since the transformation from a sawtooth wave into a triangle wave creates a reduction in harmonic richness similar (but not the same as) subtractive filters. Historically, Casio's ill-fated VZ series of synthesizers in the 1980s used a method called IPD or Interactive Phase Distortion, based on the transformation of waveforms through progressively sharper sawtooth shapes. Software glitches with the interface along with bad commercial timing (the Korg M1 released at the same time, which also had a sequencer and drums) led to the withdrawal of Casio from the pro-audio market.

With computer synthesis it is a simple procedure to create an algorithm that generates adjustable symmetry sawtooth-to-triangle waves that may be modulated at audio frequencies. Empirical research into harmonic spectra of such modulations reveals a slightly more complex morphology of spectra than would be devised using subtractive methods, and the application of single frequency modulation (sine-wave modulation) of the waveform results in complex timbre transformations over time, highly dependent on phase ratios between carrier and modulator, and a temporal morphology that reflects the characteristic shape of the sawtooth wave itself.

2. THE WAVEFOLDER~ OBJECT

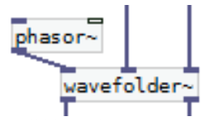


Figure 1. The wavefolder~ object generates variable asymmetry sawtooth/triangle waves from a phasor~ (ramp) input.

The implementation of an algorithm for converting a ramp wave into a triangle wave or inverse ramp is relatively simple. This was initially accomplished as a Pure Data[1] (Pd) patch using the sigpack~ library of objects[1]. More recently this has been created as an external for Pd, along with the wavestretcher~ object. This has simplified the process of converting a ramp from a phasor~ object into an adjustable-symmetry waveform, and opened-up the possibility of audio frequency modulation of the waveform symmetry.

The principle is simple. With a ramp waveform from 0 to 1, a threshold is set between 0 and 1. Sample-by-sample the output is given by:

$$O = IF(R > T; R(1/T); 1-((R-T)*(1/(1-T)) \quad (1)$$

where O = out sample, R = ramp input and T = threshold. Divide-by-zero errors are eliminated in a separate function that prevents R from arriving at precisely 0 or 1. This object can be found in the ekext library of Pd externals[2].

3. SPECTRAL CHARACTERISTICS

3.1 Frequency spectra at static symmetry settings

As the waveform is modulated between a setting of 0 (symmetric triangle waveform) and 1 (asymmetric ramp waveform), peaks and troughs in the harmonic spectrum are developed (see figures 1-4). This was empirically tested in order to establish the relationship between the symmetry of the waveform and the resultant harmonic spectrum, in order to establish how the functional de-

[1] https://puredata.info/downloads/sigpack

[2] Latest versions can be downloaded from http://sharktracks.co.uk/html/software.html

scription of a sawtooth or ramp wave is affected by this process. It makes sense to define this relationship in terms of deviation from the sawtooth or ramp waveform toward the triangle, as there is a reciprocal relationship between the troughs in the resultant spectra and the symmetry of the waveform. Furthermore, the reduction in the magnitude of even harmonics is not linear.

There is a modulation between the Fourier series of a sawtooth wave:

$$x_{saw} = \frac{1}{2} - \frac{1}{\pi}\sum_{k=1}^{\infty}\left(\frac{\sin(2\pi\, kft)}{k}\right) \quad (2)$$

and that of a triangle wave:

$$x_{tri} = \frac{8}{\pi^2}\sum_{k=0}^{\infty}\left(\frac{\sin(2\pi(2k+1)ft)}{(2k+1)^2}\right) \quad (3)$$

As can be seen from the figures below, the modulation of the magnitudes of harmonics closely resembles a cosine function of the magnitudes based on the harmonic number, starting at infinity for the ideal saw and starting at harmonic 2 for the triangle. Given that, in additive synthesis both waveforms' harmonics are alternately opposite in phase to the previous harmonic (1, -2, 3, -4 etc and 1, -3, 5, -7 etc) there are clues to how the combination of additive sine elements with different phase relationships may result in the spectra observed below. An exponential relationship between the linear asymmetry and the position of the first trough in the spectrum is observed, and the interval in harmonics until the next of these, such that a triangle wave has an absence of even harmonics (2, 4, 6, 8...interval=2) and figures for alternative symmetry settings as shown in the table and graphical figures below:

Asymmetry	Interval
0 (triangle)	2
0.5	4
0.75	8
0.875	16
0.9325	32
1 (sawtooth)	Nyquist (SR/2)

Table 1. Asymmetry settings and their correspondent troughs in the harmonic spectrum.

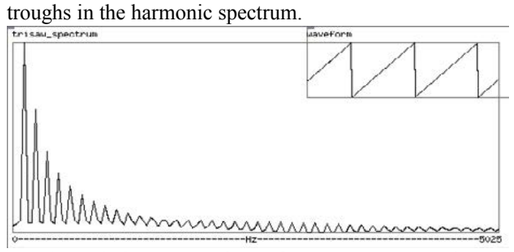


Figure 2. Spectrum and waveform at symmetry setting 1 (ramp waveform).

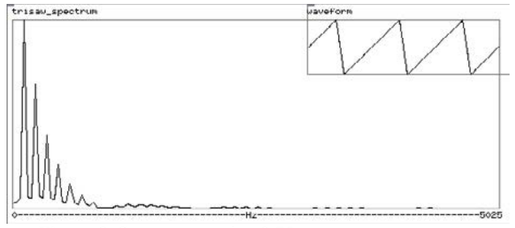


Figure 3. Symmetry setting 0.75.

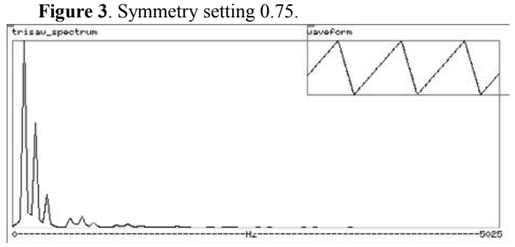


Figure 4. Spectrum and waveform at symmetry setting 0.5.

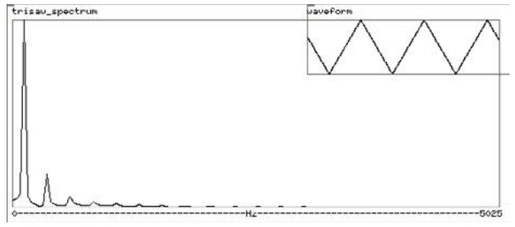


Figure 5. Spectrum and waveform at symmetry setting 0 (triangle waveform).

4. AUDIO FREQUENCY MODULATION OF THE WAVESHAPE

The shape of the wavefolder~ output is controllable at audio rate with limits of -1 (saw down) and 1 (saw up) with a setting of 0 representing the triangle waveform. The relationships between the phase of the modulation signal (in this case a simple sinusoidal waveform) and the phase of the asymmetry modulation are important to the resulting timbre. With a modulating sine function at the same frequency, at 270º there are more corners to the waveform, and more high-frequency harmonics are generated (fig. 7), whereas at 90º between trisaw and sine the waveform is more like a distended triangle wave and the harmonic spectrum is less bright (fig. 6).

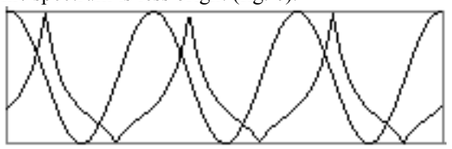


Figure 6. Superimposed waveforms of modulator and resultant waveform at modulator phase = 270º with respect to the tri/saw wave.

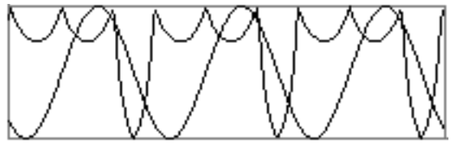


Figure 7. Superimposed waveforms of modulator and resultant waveform at phase = 90º.

4.1 Spectro-Morphology at Detuned Modulation Frequencies

Thus far this paper has considered static waveforms, and there is no single result here that cannot be achieved by a wavetable method of synthesis. But this method begins to yield more interesting results as the modulation waveform is detuned from integer-multiples of the asymmetric modulated waveform. The phase relationship discussed above is continually changing, and this results in morphological transitions between very bright, harsh-sounding timbres and softer timbres.

At a positive detuning away from the frequency of the tri/saw wave, the sweep is from bright-to-soft with a plateau at the brightest point, repeating at a rate equivalent to the difference in frequency between the carrier (tri/saw waveform) and the modulator (sine). The inverse is true at a negative detuning, that is the sweep in timbre is from soft-to-bright. More complex timbres are achieved with simple non-integer ratios (1.5, 0.75 etc) giving inharmonic timbres but with a degree of tonality. Just as with frequency modulation synthesis, the more complex the integer ratio of the carrier to the modulator, the more inharmonic the timbre produced.

Furthermore, since the sweep in brightness is a rhythmic effect, this can be controlled mathematically to be consistent across all integer-ratio carrier-to-modulator values, and an object has been created to facilitate this, which will be demonstrated at the conference and made available on the author's website.

4.2 Pulse-Width Modulations of the Modulated Asymmetric Waveform

The wavefolder~ object has an extra inlet and outlet at audio rate allowing for the modulated waveform to have a process of pulse-width modulation applied to it. Since the waveform shapes of a modulated asymmetric waveform are geometrically complex, a set of timbres are available from the object that are more varied than those of traditional PWM. When this is combined with the detuning of the modulator discussed above, the timbre evolution of the asymmetric waveform is transferred to the pulse waveform with the potential for modulations of the PWM threshold to create further evolutions in timbre.

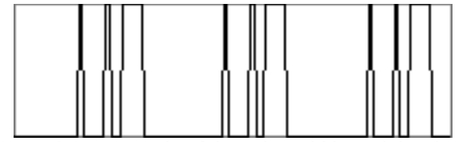


Figure 8. An example of the pulse-width modulated output from wavefolder~.

5. MORE PIECEWISE MANIPULATION

5.1 Wavestretcher~

A second object uses a similar approach the the wavefolder~ by taking a breakpoint (threshold) and manipulating the geometric angle of the waveform differently depending on which side of the threshold it is. It is useful to think of this as a complementary function to the previous object. While the wavefolder~ modulates from a sawtooth input (from phasor~) towards a triangle waveform using a breakpoint-based algorithm, wavestretcher modulates from the sawtooth (or any input waveform for that matter) towards pulse-train-style waveforms as shown below.

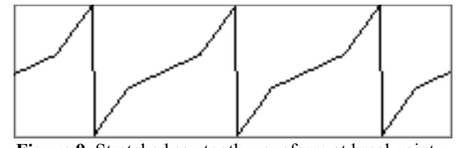


Figure 9. Stretched sawtooth waveform at breakpoint = 0 (middle of absolute value) and stretch factor at -0.5.

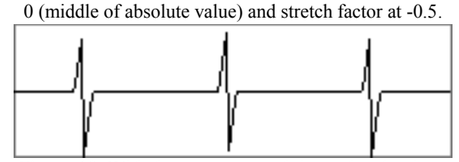


Figure 10. With the same sawtooth input, breakpoint = -0.75, stretch factor = -1.

Positive values of the stretch factor allow the modulation between triangular or sawtooth waveforms through trapezoidal waveforms until a square wave or clipped sawtooth waveform results.

Figure 11. With the same sawtooth input, breakpoint = -0.5, stretch factor = 0.7.

The use of both objects, with the output of wavefolder~ feeding into wavestretcher~ affords a situation where a large repertoire of complex timbres may be generate using a highly compact, efficient structure. It is possible to emulate timbres of subtractive synthesis without the use of filters, but with a greater degree of flexibility in terms of timbre control[3].

[3] It must be stated that there is no way to reproduce high-Q resonant peaks without the use of audio filters in this system, although a phase-distortion-type equivalent may involve added resonant circuits to one portion of the waveform.

6. ANALOG REALIZATION

6.1 Sawtooth to Triangle Wave Modulation

There are some implementations of this idea available on synth-DIY sites by Hoshuyama [2]. Tillmans [3] and Gratz [4]. All of these articles observed by this author come with the caveat that they are "untested" e.g.[2] although this seems unlikely given the knowledge and experience of those contributing this knowledge, since such features exist in Moog and MFB synthesizers (Moog Voyager, MFB Dominion) and it would be naive to assume that the potential of these systems was overlooked[4], especially given the ubiquity of PWM in commercially successful forms of popular electronic music and electronic dance music (EDM) over the past four decades. However only the Moog Voyager XL appears to offer a fully patchable (and hence audio rate) modulation of the waveform shape.

Of particular interest for its simplicity is the design and article by Don Tillmans [2], published in 2000 and revised in 2002, providing a simple circuit for analog waveshaping of a sawtooth wave using two operational transconductance amplifiers. A certain amount is left to the circuit-builder to figure out in this article. As the original circuit uses hard-to-find CA3280 chips[5] efforts are ongoing to adapt the circuit to use a readily available LM13700 dual operational transconductance amplifier (OTA) integrated circuit (IC), although a new solution is detailed below, based on the wavefolder~ algorithm.

The analog circuit designed by Don Tillmans (cited by Gratz[3]) uses an equivalent equation to that in the digital object wavefolder~ expressed in a form more mathematically elegant than the coding algorithm expressed in part 2 above, thus:

$$\frac{1}{1+e^{x}}+\frac{1}{1+e^{-x}}=1 \qquad (4)$$

where x is equivalent to the control voltage in the analog circuit, and the threshold/breakpoint value in the digital algorithm of wavefolder~. This accurately reflects the exponential relationship between the wavefolder~ threshold value and the harmonic modulations detailed in table 1, and figures 2-5.

Given that an exponential multiplication of a signal is the reciprocal of a division by a linear increase or decrease of the denominator, an alternative method can be devised for an analog circuit using the principles of the original wavefolder~ algorithm. An analog switch IC replaces the IF statements, and differential amplifiers are used to generate reciprocal control voltages for a voltage-controlled amplifier against a reference voltage. Both the input ramp wave and its inverted counterpart are switched alternately using a comparator, along with the control voltages to an exponential converter into an OTA. A single OTA may be used, and in this realization it is a CA3080 – the chip designed by RCA that was vital to the creation of early voltage-controlled synthesizers (the RCA Mk1 and Mk2) which, as the footnote below shows is now available again, albeit in lots of 100 ICs.

This circuit should perform in exactly the same way as the digital algorithm, with a threshold voltage determining the asymmetry of the resultant waveform. Effectively though, every analog implementation of this principle, from the low-frequency oscillator of the Korg MS20 to the switched OTA concept described above, uses the same principle of threshold-switching the separately amplified non-inverted and inverted portions of the ramp waveform on either side of the switching threshold.

[4] These articles are all over a decade old at the time of publication.

[5] These are now being manufactured by Rochester Electronics: http://www.rocelec.com

7. CONCLUSIONS

This project was driven by curiosity into a way of generating complex timbres from simple means, and how pushing methods from analog experimentation by synthesis enthusiasts into the digital domain may open new approaches (asymmetry modulation) to timbre modulation of basic synthesis waveforms. The conceptual process of development for the analog circuit was realized by understanding that through some lateral transposition of the principles of the digital implementation, an analog realization could be created based on the same principles as the digital object. A conceptual loop can be observed where code-based digital methods and analog electronics can be created in parallel, and where understanding from one branch of electronic music can be adapted to function using the same principles in another.

Acknowledgments

This project was devised and researched by the author, and I am deeply grateful to the University of the Arts London, and particularly Nick Gorse and Jonathan Kearney for releasing funding for its presentation. A great deal of credit for understanding analog circuits in voltage-controlled synthesis should be given to Thomas Henry, Ray Wilson, Ian Fritz and Don Tillmans.

8. REFERENCES

[1] Puckette, M, *Pure Data: another integrated computer music environment,* Proceedings, Second Intercollege Computer Music Concerts, Tachikawa, Japan, pp. 37-41, 1996

[2] Hoshuyama, O, *Wave-Shaper (Variable Slope-Ratio Triangular),* http://www5b.biglobe.ne.jp/~houshu/synth/WvShp0306.gif, 2003.

[3] Tillmans, D, *Voltage Controlled Duty Cycle Sawtooth Circuit,* www.till.com, 1999, 2002.

[4] Gratz, A, *Triangle / Sawtooth VCO with voltage controlled continously variable symmetry,* http://synth.stromeko.net/diy/SawWM.pdf, 2006.

Towards an Aesthetic of Instrumental Plausibility for Mixed Electronic Music

Richard Dudas
CREAMA
Hanyang University
Seoul, Korea
dudas@hanyang.ac.kr

Pete Furniss
Reid School of Music
Edinburgh College of Art
University of Edinburgh, UK
p.furniss@ed.ac.uk

ABSTRACT

The implementation of live audio transformations in mixed electronic music raises the issue of plausibility *in real-time instrumental transposition. The composer-performer collaboration described in this paper deals with two of the composer's existing pieces for solo instrument and computer, addressing issues of timbre and intonation in the output and adapting the existing software with improvements informed by both the physical resonating properties of musical instruments and by instrumental ensemble practice. In preparing these pieces for publication, wider performance and further instrumental transcription, improvements stemming from both compositional and performative considerations were implemented to address this issue of plausibility. While not attempting to closely simulate a human approach, the authors worked towards a pragmatic heuristic that draws on human musical nuancing in concert practice. Alternative control options for a range of concert spaces were also implemented, including the configuration of user input and output at interface level in order to manage common performance-related contingencies.*

1. INTRODUCTION

Following several years of collaboration on the performance of mixed electronic music, the authors decided to return to two existing pieces, *Prelude I for Clarinet and Computer* and *Prelude II for Clarinet and Computer*, in order to modify and update their technological component. The primary motivation behind this was simply to make the audio processing "sound better" for the purposes of including them on a published sound recording. The second motivation was to prepare the pieces for wider dissemination, including transcription of one of the pieces for a variety of other instruments, through publication of the score and technical materials. Both motivations necessitated an updating and refinement of the underlying audio processing. Improvements to the audio signal processing were geared toward the implementation of a *plausible* instrumental transposition – one that is informed by physical resonating properties of instruments and by instrumental ensemble practice. In doing this, we were not attempting to simulate a human approach, but rather create something pragmatic that draws on human musical nuancing.

The desire for a plausible instrumental transposition required addressing the way that audio effects can modify the perception of instrumental resonance in uneven ways. While resonant filter banks have been used frequently to simulate instrumental resonance where sound synthesis is concerned [1, 2], here they were employed to provide a homogeneity within the transposed material. Furthermore, the recent move to 64-bit has brought subtle improvements to clarity in audio signal processing that become musically significant in multi-layered musical and sonic textures. It was therefore necessary to make updates at the code level to some project-specific software. Finally, it was decided to break from an exclusive use of equally tempered semitones as a subtle step in an attempt to impart a chamber music aesthetic to the computer processed output.

The work undertaken on instrumental transcription and interface design represents a continuation of the authors' earlier research on *Prelude I*. [3] Improvements to the interface and audio processing chain were implemented in order to address the configuration of user input and output for the management of common contingencies in the performance space. The existing user interface was further adapted to provide a diversity of options for control either onstage by the performer or offstage by a technical assistant.

2. RESONANCE AND FORMANT FILTERING

The two pieces referred to here each employ real-time transformations of the live input, including transposition both within and beyond the actual range of the instrument(s) to create the effect of a virtual ensemble. Where these transpositions extend beyond a perfect fifth in either direction, the question of plausibility becomes an issue [4] in respect of an overall ensemble aesthetic. In the case of the *Prelude I*, transcription of the original solo flute part to both violin and viola had led the composer to adapt the software to string instruments by first filtering out the fixed formant structure of the resonating instrumental body using notch filters before transposition, and later adding the formants back into the transposed sound through the use of resonant filtering. This creates a greater sense of homogeneity, since the formants remain stable when the sound is transposed. *(Refer to Sound Example Set 1 via the link provided at the end of this paper.)*

In comparison to stringed instruments, the spectral envelope of woodwind instruments is heavily dependent on both pitch and volume (this is especially true of the clar-

inet and flute), and does not have an entirely fixed formant structure. However, a generalized pitch-invariant formant-based model has been shown to be helpful in improving perceptual evaluation of synthesized instrumental tones [1, 2], so the technique used with the string versions of the piece seemed potentially appropriate for any woodwind transcription. Therefore, in addition to including formant filtering in the violin and viola transcriptions of *Prelude I*, it was decided to introduce a similar filtering system to the clarinet version of the piece [3], in order to improve the instrumental perception of transposed audio material. [1] [5] Had string versions of this piece not been created, the use of fixed formant filtering would probably not have been considered for wind instrument transcriptions, however, since the filtering had been implemented in the performance software and proved to be effective with the clarinet when tested empirically, it was added to the clarinet version.

Wind instruments are inconsistent in their resonating tube lengths across the range of musical pitches available, in comparison to other instrument classes. [6, 7] Orchestral stringed instruments, the guitar and the majority of percussion, for example, maintain much the same resonating body over a variety of frequencies. In valved brass instruments, there are a small number of differing resonating tube lengths, according to which valves (or combinations thereof) are depressed, or which harmonic is being emphasised by the embouchure of the player. The trombone has a relatively consistent, though smoothly scalable resonating tube. In order to assimilate a plausible overall resonance for the clarinet, we decided to take samples from each part of the instrument's range using *glissandi* in order to find the prominent stable resonant frequencies in the spectrum, which could be filtered out before transposition and added again afterward. This is consistent with the successful implementations in the string versions of the piece: the resulting transpositions become more credible in terms of the instrumental textures they are modeling. *(Refer to Sound Example Set 2 via the link provided at the end of this paper.)*

3. UPGRADING DIGITAL AUDIO RESOLUTION

Upgrading the software to 64-bit sample compatibility was initially enacted out of the necessity to maintain performance software. [3] However, side-by-side comparisons of 32-bit and 64-bit audio output were also found to present higher clarity and definition, particularly noticeable in musically dense passages. [2] *(Refer to Sound Example Set 3 via the link provided at the end of this paper.)* On a technical front, the update of the MSP external objects themselves was fairly straightforward: it simply required making minor modifications to the code in accordance with the specifications in the most recent Max Software Developer's Kit and recompiling.

A sample rate hike to 48kHz or much higher was discussed, which could potentially further improve the quality of transposed sounds from the live input. However, although the recording industry currently leans towards 96/192kHz, such a retrofit would have required considerable recoding of the patch, which was considered overly laborious to be of notable benefit at the present time. Nevertheless, higher rates will certainly be investigated for future pieces in this series.

4. INTONATION

> *"The question of intonation is evidently relevant to nearly all instrumentalists ... and has a profound influence on the way composers and performers collectively think about harmony and intonation."* – Mieko Kanno

Where real-time transpositions are concerned, manually hard-coding intonation choices is burdensome and time-consuming, even with the aid of the computer to help pre-calculate ratios to semitones; this is especially true when compared to the immediate and multiplex tuning adjustments that trained musicians make by ear. [8, 9, 10, 11, 12, 13] Therefore it was decided to create an algorithm to automate intonation for real-time transposed chordal structures defined in the score in terms of semitones for these pieces, in lieu of simply providing transposition in cents, or fractions thereof. This is not an issue of user interaction but rather a compositional and aesthetic choice. There are prior examples of this kind of system, such as Eivind Groven's automat for adaptive just intonation [14, 15], the algorithm for which makes a note choice from a selection of fixed pitches. Many such systems are appropriate for keyboard instruments (Groven's was first implemented for organ), but the basis for these is quite different from the type of tuning that other instrumentalists or vocalists may intuitively execute in performance.

Kanno cites Fyk in relation to four distinctive types of "expressive tuning" employed by instrumental musicians (and singers): "harmonic, melodic, corrective and colouristic." Harmonic tuning relates to just intonation in relation to explicit or implicit vertical structures, while melodic intonation concerns a relative broadening and tightening of intervals based on melodic direction. Corrective intonation is "instinctive tuning" which occurs when a performer hears a discrepancy between projected and perceived pitches, while fine adjustments of timbre may be achieved by colouristic intonation choices. All of these ongoing manipulations of pitch require "the linearity of time against which to map out [their] expressive intention." [10]

Many common chords (such as triads or secundal/quartal harmonies) are relatively straightforward to tune. For example, a major triad consists of three justly tuned intervals: a perfect fifth (3:2 ratio or 1.5) defined by the outer pitches, a major third (5:4 ratio or 1.25) and a minor third (6:5 ratio or 1.2), both delineated by the central pitch?s relation to the two outer pitches. Each of the just intervals aligns perfectly with the others to form the triad (1.25 * 1.2 = 1.5). This would not be the case for a three-note vertical structure with a sharp 4th degree such as C F# G, since the just ratios for the tritone (11:8 or 1.375) and minor second (17:16 or 1.0625) do not superimpose to comprise a justly tuned perfect fifth (1.375 * 1.0625 = 1.4609375 not 1.5). [3] This implies that, if we want the outer notes to delineate a justly tuned perfect fifth, we need to make a choice between tuning the F# to the C (with a just tritone), or tuning it to the G (with a just minor second), since it cannot be justly tuned to both notes and yield an in-tune perfect fifth between the outer notes. In each of scenarios the F# will cause beating with one of the two other notes. Alternatively, we could split the difference between tritone and minor second and tune the central note somewhere in-between, so it beats evenly (or even unevenly) with both of the (in tune) outer notes. For many musicians, including string players but also singers and players of wind or brass instruments, the fine adjustment of this intonation is an instinctive, internalized process based on years of experience and deliberate practice. For the computer, however, it is rather more complicated, since the programmer must create an algorithm to find the appropriate tuning nuance in each case.

The algorithm used here measures the intervallic content of chordal structures, calculates individual frequency ratios for each interval, and adjusts the calculated frequency of each note based on the weighted consonance of each interval within the chord [16, 17, 18], with respect to a reference pitch in the chord (usually the note being played by the live instrumentalist in mixed electroacoustic music). Once a midi note is identified from the input, it is converted to a frequency value and thereafter dealt with on a ratio basis instead of using semitones and cents. A very slight amount of random variation proportional to the frequency of any given pitch is added to the final tuning to emulate human error, and keep highly justly tuned chords from becoming static, seeming too mechanical or perceptually fusing into a single note. This humanizing of the intonation is heuristically modeled on the various types of tuning that string players perform for double-stops, or that the individual performers in chamber or vocal ensembles (without piano) perform when tuning chords. [10, 11] The outcome is that such a system can be used in future works, or retrospectively implemented in other compositions in which a system of flexible, unequal-tempered tuning improves the *instrumental plausibility* of the electronic effects used in that composition. As always, the objective is not to attempt to faithfully simulate a human approach, but rather create a pragmatic method inspired by human performance that draws on musical nuancing. *(Refer to Sound Example Set 4 via the link provided at the end of this paper.)*

5. INPUT AND OUTPUT

The software was restructured to allow for various user-configurable inputs, as well as gain and balance controls at various stages in the processing path. This is often overlooked, but allows the patch to be tailored to a variety of different performance scenarios. Similar one-patch-fits-all approaches have been used by a variety of composers (e.g., Kaija Saariaho, Martin Parker and Alexander Harker among others). It was also noted here that an increasing number of performers in the field are preferring to run their electronic parts directly, without the aid of an offstage technician. These user-oriented input and output controls within the onscreen interface allow for relatively rapid and simple adjustments to be made by the performer *in situ*, in response to aspects of the performance space, such as balancing microphone signals into the system, setting live direct output level and managing feedback with a variety of input solutions.

While many clarinetists employ two or more microphones to adequately cover the range of the instrument, an additional, isolated (eg. piezo or contact microphone) input may be used to track pitch without feedback from the software output. [3] Although the quality of these transducers may be inferior to a high quality condenser or ribbon microphone [4], it was considered advantageous to provide for the use of up to three inputs for any instrument, with the ability to mix relative levels directly from the user level of the software interface. In our case, the third input was disregarded entirely for output purposes (being used only to feed the pitch tracker). This allows a performer to make adjustments quickly according to varying performance ecologies (the acoustic space, loudspeakers, microphones etc), which may differ considerably to rehearsal conditions, in order to manage feedback and projection levels. *(Refer to Sound Example Set 5 via the link provided at the end of this paper.)* A continuation of this input/output configurability relies on performers adopting strategies such as presets, VST plug-ins, or drop-ins [19], in order to assert established priorities regarding their overall sound.

The restructuring also involved separating the sound sources produced in the piece from a fixed speaker definition, so it can be performed with any given multichannel (or stereo) speaker configuration. It was previously limited to 4 or 2 channel output – it now deals with spatial location using a standard azimuth definition [5] – presuming that the speakers are arranged more or less along a circular path around the hall, when using a multichannel setup. (In a stereo scenario the left-right panning information is extracted from the azimuth.)

6. CONCLUSIONS

In implementing the above improvements to two existing pieces, it was discovered that the additional filtering before and after transposition was worth the effort in terms of a timbral strengthening of the instrumental plausibility in the electronic part. Furthermore, although the upgrade to 64-bit signal processing was dictated by the necessity of software maintenance, it was a pertinent element in improving the clarity of the electronics, particularly in densely scored sections.

Further research to determine the perceptual effectiveness of intonation adjustments could be undertaken using a music perception experiment, with the results used to determine improvements to the algorithm. In the two pieces discussed in this paper, both instrumental plausibility and

[1] It was not necessary to revisit the original flute version since the basic spectral correction used for that version was already taken into into consideration during the compositional process.

[2] For a single stream of audio processing there is often little to no perceptual difference between the two, however there *is* an audible difference when mixing multiple audio streams. This could have something to do with dither being at a lower volume when mixing multiple sources, or just an increased amplitude resolution mitigating high frequency phase cancellation when mixing multiple signals.

[3] The same holds true if other ratios such as 7:5, 10:7, 24:17, 45:32, etc. are used to define the tritone.

[4] Considerable improvements continue to be made in this area. For example, we use the Rumberger K1X Advanced Piezo Technology condenser microphone mounted within a specially adapted clarinet barrel. The relatively high quality of this device gives further options in the blending of input sources to accurately reflect the performer's sense of priorities in terms of their sound.

[5] Vector based panning (VBAP) [20] was used to accomplish this.

effective transcription [3] were enabled by the adaptation of filtering and an approach informed by both the physical properties of instruments and the needs and priorities of their players within a variety of performance environments.

Acknowledgments

This research was part-funded by the Arts and Humanities Research Council, UK (AHRC). The authors are also grateful to hosts Hanyang University College of Music and the University of Edinburgh (Edinburgh College of Art). Thanks also to Dr. Miriam Akkermann, for her input and assistance during stages of this research.

7. URL

http://www.richarddudas.com/ICMC2016Sounds/

8. REFERENCES

[1] S.-A. Lembke and S. McAdams, "A spectral-envelope synthesis model to study perceptual blend between wind instruments," in *Proceedings of the Acoustics 2012 Conference*, Nantes, 2012, pp. 1031–1036.

[2] ——, "The Role of Spectral-Envelope Characteristics in Perceptual Blending of Wind-Instrument Sounds," *Acta Acustica united with Acustica*, vol. 101, pp. 1039–1051, 2015.

[3] P. Furniss and R. Dudas, "Transcription, Adaptation and Maintenance in Live Electronic Performance with Acoustic Instruments," in *Proceedings of the Joint International Computer Music Conference and Sound and Music Computing Conference*, Athens, 2014, pp. 456–462.

[4] S. Bernsee. Time Stretching And Pitch Shifting of Audio Signals – An Overview. [Online]. Available: http://blogs.zynaptiq.com/bernsee/time-pitch-overview/

[5] R. Dudas, "Spectral Envelope Correction for Real-Time Transposition: Proposal of a 'Floating-Formant' Method," in *Proceedings of the International Computer Music Conference*, Gothenburg, 2002, pp. 126–129.

[6] A. H. Benade, *Horns, Strings, and Harmony*. Dover Publications, 1992[1960].

[7] N. Fletcher and T. Rossing, *The Physics of Musical Instruments, 2nd Edition*. Springer Verlag, 1998.

[8] K. Sassmannshaus. [Online]. Available: http://www.violinmasterclass.com/en/masterclasses/intonation

[9] R. W. Duffin, *How Equal Temperament Ruined Harmony (and Why You Should Care)*. W. W. Norton and Company, 2007.

[10] M. Kanno, "Thoughts on How to Play in Tune: Pitch and Intonation," *Contemporary Music Review*, vol. 22, no. 1-2, pp. 35–52, 2003.

[11] S. Fischer, "Intonation," *The Strad*, vol. 125, no. 1495, pp. 78–79, 2014.

[12] ——, *Basics*. London, UK: Peters Edition Ltd., 1997.

[13] L. Auer, *Violin Playing as I Teach It*. Toronto: Dover Publications, 1980[1921].

[14] J. Rudi, "Eivind Groven's automat for adaptive just intonation: A pioneering example of musically situated technology," *Studia Musicologica Norvegica*, vol. 41, pp. 40–64, 2015.

[15] ——, "The Just Intonation Automat – a Musically Adaptive Interface," in *Proceedings of the International Computer Music Conference*, Denton, 2015, pp. 42–45.

[16] A. Plomp and J. L. Levelt, "Tonal Consonance and Critical Bandwidth," *Journal of the Acoustical Society of America*, vol. 38, no. 4, pp. 548–560, 1965.

[17] A. Kameoka and M. Kuriyagawa, "Consonance theory, part I: Consonance of dyads," *Journal of the Acoustical Society of America*, vol. 45, no. 6, pp. 1452–1459, 1969.

[18] ——, "Consonance theory, part II: Consonance of complex tones and its calculation method," *Journal of the Acoustical Society of America*, vol. 45, no. 6, pp. 1460–1469, 1969.

[19] M. Edwards. input-strip. Open source Max/MSP software. [Online]. Available: http://www.michael-edwards.org/software/

[20] V. Pulkki, "Compensating displacement of amplitude-panned virtual sources," in *Audio Engineering Society 22th Int. Conf. on Virtual, Synthetic and Entertainment Audio*, Espoo, 2002, pp. 186–195.

Noise in the Clouds

Eva Sjuve
University of Huddersfield
School of Music, Media and Humanities
eva.sjuve@hud.ac.uk

ABSTRACT

This poster describes Metopia, a research project in music composition, which consists of creating compositions for the sky and the clouds, with a wireless sensor network to sense the state of the air in urban spaces. This sensor network acquires a complex set of data from the air for the purpose of making a set of sound compositions, using the programming language Pure Data in combination with embedded computers. This research project is using a real-world problem such as air-pollution as a way to explore a responsive sky to communicate the state of the toxic level into a real time auditory response. Atmospheric pollutants is a major health issue and Metopia is one way of examining this problem through aesthetic and conceptual choices using generative principles. Noise in the air from toxic substances is examined through the aesthetic choices in a music composition.

1. INTRODUCTION

Metopia is part of research in music composition using generative principles for composing from unstable data, such as the complex process of circulating air, to communicate this noise in the sky through aesthetic principles. How can toxic noise in the sky be sonified in a music composition?

1.1 The Black Sky - Air Pollution

Toxic substances in the air is a major health problem. The World Health Organization (WHO) attributed 3.7 million deaths due to ambient pollution in 2012 [1]. Air pollution measurements show that European cities have high measurements of toxic substances, a demanding health challenge. Nitrogen Dioxide coming from Diesel vehicles, with London having the highest concentration in Europe [2] and 40 000 deaths each year are attributed to air pollution [3]. Weather sensors are all around us. Airplanes use sensors to collect data about storms and turbulence, and delays in air traffic cost airlines $8 billion each year world wide [4].

1.2 Particulate Matter in the Clouds

Particulate matter are detected in Metopia using various sensors, such as electro chemical, Volatile compound sensors, and dust sensors. What we humans breathe is the focus of this composition, particulate matter in the clouds. clouds. In Metopia (meta + topos, the place beyond) the air pollution in the environment is examined using a scalable wireless sensor network, measuring amongst other data, particulate matter, which is respirable dust, particles smaller than 5 microns that have the possibility to enter the gas exchange region of the lungs [5]. One early prototype of one of the nodes in the wireless sensor can be seen below in Figure 1.

Figure 1. Metopia, an early prototype with Raspberry Pi 2, Arduino, and an array of sensors.

2. CLOUD MUSIC

Using measurements of environmental data or weather data for music compositions or sonification is nothing new and many projects has been explored this in composition and sonification, and to mention a few are Marty Quinn's The Climate Symphony [6] and James Bulley *et al.*[7].

An early work taking the starting point in atmospheric variations, the movements of clouds in the sky and using the data for timbral variations in a musical real-time composition was *Cloud Music*, a sound installation

by Robert Watts, David Behrman and Bob Diamond, exhibited between 1974-1979. The movement of clouds in the sky is controlling the music of a synthesizer. In the installation a black and white video camera is pointing to the sky with six crosshairs on the display, to make music of the movement of clouds. The system analyzed the light variations in the video image from the six data points, and sends this data into a custom built synthesizer as control voltage, to converts the changes into harmonic progressions and dynamic shifts in the music [8]. See the design of the system of Cloud Music in Figure 2 below.

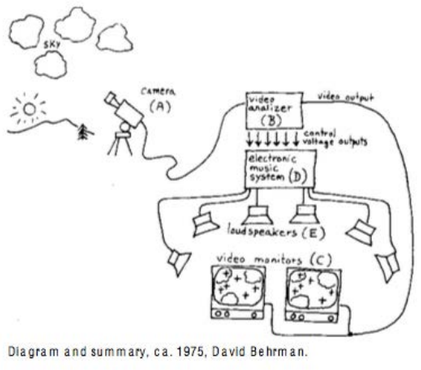


Figure 2. Cloud Music from 1975. Image courtesy David Behrman.

3. SONIFICATION OF THE SKY

Air pollution and the acquired data is in this composition used as a generative principle in the composition. The sky is acting as material in the musical process and its indeterminate variation is shaping the composition both on a macro-level and on a micro-level. The data acquisition from the sensors is used in forming the generative system in Metopia, using rule-based principles of circadian waves, short waves and bursts explored in a previous paper publication [9].

3.1 Metopia System Design

The data acquisition is using an array of sensors and the data is sonified in real-time using a generative system. Metopia's system consists of environmental sensors measuring air pressure, temperature, light, volatile gases and dust. One sensor is sensing the electro-magnetic propagation in the machine [9]. The data is acquired by a micro-controller, an Arduino [11] in this prototype, connected to a Linux/GNU Debian operating system on a Raspberry Pi 2 [12], where Pure Data is processing the data[13]. The mesh network is made up of the Xbee low power modules, using the ZigBee protocol DigiMesh for communication [14]. As this is a work-in-progress, a final composition is not described here, but only one part of it.

3.2 Generative System

The sound processing is made by a generative rule-based engine, connecting the physical dimension to the auditory dimension through a processing engine. No sample-based sonic material is used in this composition, only low-level synthesis. The generative system is described here briefly.

The different dynamics of the system is conceptualized as long waves, short waves and bursts and these dynamics have a direct relation to timbral qualities and musical gestures in the composition, and is in direct relation to the variation in data acquisition from the physical world. The earth system with its temporal variations, with its long atmospheric tides are treated as long waves in this composition. The atmospheric sensor reading conceptualized as long waves, such as data from air pressure, is in this composition controlling the fundamental frequency of the harmonics. Volatile gases have the temporal characteristic of short waves in the physical dimension and is treated as such in the composition where they modulate timbral variations and musical gestures. The electro-magnetic propagation has the characteristics of burst of noise and control rhythmic patterns and short musical gestures.

4. NOISE IN THE CLOUDS – NOISE IN THE SYSTEM

Noise in this project Metopia is not only referring to air pollution as a musical material, but also as a musical material, such as electro-magnetic propagation, that traverses the sky as atmospheric disturbances and in the electrical system, the circuit of the machine. In this semi-automatic system, noise can come from various sources in the system, such as wires or onboard components, the power circuit, or intermodulation which makes the system unpredictable [15]. This unpredictable noise is part of the aesthetics of the system's generative principles, including both unstable data and the unstable sky and is explored in the composition as noise as a music material. The sky is also affected by the unpredictable earth system, such as volcanos with particulate matter traveling long distances around the globe. See the eruption of the Eyjafjallajökull in 2010 in the Figure 3 below, as an example of the sky as an unpredictable system.

Figure 3. Eruption of the Eyjafjallajökull Volcano on Iceland putting air traffic on a hold in 2010.

5. DISCUSSION

The use of data gathered from sensors is focusing on an unpredictable system to create a generative system. The aesthetics of noise in the clouds and noise in the system is explored to sonify the unpredictable variation within the earth system. This research project is still a work in progress and at this stage it is too early to draw any conclusion of the generative composition in relation to data acquisition, but as a system for creating compositions of complex data it has potential for further explorations in principles for generative music.

2. REFERENCES

[1] World health organization (WHO). 2012 [online] <http://www.who.int/gho/phe/en/> *(Accessed January 20, 2016)*

[2] B. Johnson, The Guardian, 25th of February, 2016 [Online] <http://www.theguardian.com/environment/commentisfree/2016/feb/23/the-guardian-view-on-air-pollution-breathe-uneasy> *(Accessed February 20, 2016)*

[3] Royal College of Physicians, [Online] <https://www.rcplondon.ac.uk/projects/outputs/every-breath-we-take-lifelong-impact-air-pollution> *(Accessed January 28, 2016)*

[4] N. Goyal, Industry Tap, [Online] <http://www.industrytap.com/weather-sensors-turn-airplanes-meteorologists/21735> *(Accessed January 28, 2016)*

[5] Engineering Toolbox [Online] < http://www.engineeringtoolbox.com/particle-sizes-d_934.html> *(Accessed January 28, 2016)*

[6] M. Quinn, "Research Set to Music: The Climate Symphony and Other Sonifications of Ice Core, Radar, DNA, Seismic and Solar Wind Data" in *Proceedings of the 2001 International Conference on Auditory Display, Finland, 2001*

[7] Bulley, J, Jones, W, "Variable 4: A Dynamic Composition for Weather Systems", in *Proceedings of the International Computer Music Conference 2011*, Huddersfield, 2011

[8] D. Behrman, [Online] *<http://www.dbehrman.net/Collaborations/cloud_m132.html> (Accessed February 10, 2016)*

[9] E. Sjuve, "Noise in the Clouds: Noise in the System". In *Proceedings International Symposium of Electronic Arts 2016*, City University of Hong Kong, 2016.

[10] Arduino Uno Rev.3 [online] <http:// http://www.arduino.cc> *(Accessed February 12, 2016)*

[11] Raspberry Pi 2. [online] <http://www.raspberrypi.org>

[12] Pure Data. [online] < http://www.puredata.info> *(Accessed February 2, 2016)*

[13] ZigBee DigiMesh. [online] < http://www.digi.com/products/xbee-rf-solutions/modules/xbee-pro-900hp#specifications> *(Accessed January 22, 2016)*

[14] *Pure Data. [online] < http://www.puredata.info>(Accessed January 20, 2016)*

[15] P. Horowitz, and W. Hill, "*The art of electronics"*. Second edition. Cambridge: Cambridge University Press 1989, pp. 515

Musical Style Modification as an Optimization Problem

Frank Zalkow
Saarland University
frank.zalkow@uni-saarland.de

Stephan Brand
SAP SE
stephan.brand@sap.com

Bejamin Graf
SAP SE
benjamin.graf@sap.com

ABSTRACT

This paper concerns musical style modification on symbolic level. It introduces a new, flexible method for changing a given piece of music so that its style is modified to another one that previously has been learned from a corpus of musical pieces. Mainly this is an optimization task with the music's note events being optimized for different objectives relating to that corpus. The method has been developed for the use case of pushing existing monophonic pieces of music closer to the style of the outstanding electric bass guitar player Jaco Pastorius.

1. INTRODUCTION

Musical style is a quite vague term that is at risk not to be captured computationally or even analytically. The musicologist Guido Adler states in his early, grand monograph about musical style:

> So [regarding the definition of style] one has to content oneself with periphrases. Style is the center of artistic approaching and conceiving, it proves itself, as Goethe says, as a source of knowledge about deep truth of life, rather than mere sensory observation and replication. [1, p. 5] [1]

This passage suggests not to approach musical style computationally. Nearly 80 years after that Leonard B. Meyer expresses a rather opposite view:

> Once a musical style has become part of the habit responses of composers, performers, and practiced listeners it may be regarded as a complex system of probabilities. That musical styles are internalized probability systems is demonstrated by the rules of musical grammar and syntax found in textbooks on harmony, counterpoint, and theory in general. [...] For example, we are told that in the tonal harmonic system of Western music the tonic chord is most often followed by the dominant, frequently by the subdominant, sometimes by the submediant, and so forth. [2, p. 414]

There are a couple of things to notice here: First, Meyer supports the view that style isn't in the music *per se*, but only when regarded in relation with other systems. Second, style is seen probabilistic, supporting the attempt of this paper to tackle style computationally. Third, in the textbooks he mentions, probabilities are used in a very broad sense. Words like *frequently* or *sometimes* aren't enough for the models of this paper. So one cannot rely on textbooks and has to work through real data.

Working through data is the guidance of this new approach for musical style modification, i.e. changing a given piece of music so that its style is modified to another one that previously has been learned from a data corpus, while the original piece of music should shine through the new one (section 2). This style modification is seen as an optimization problem, where the music is to be optimized regarding different objectives (section 4). The method has been developed for monophonic melodic bass lines, along with chord annotations—especially for the style of the outstanding electric bass guitar player Jaco Pastorius (see section 5 for some results). Due to the nature of the use case, the method is oriented towards monophonic symbolic music, but most parts of the procedure could easily be extended to polyphonic music as well. In this case the chord annotations may be even computed in a preceding automated step, so that annotating would not be necessary.

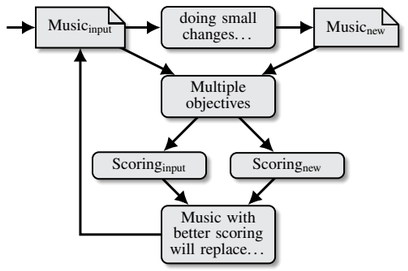


Figure 1: Rough overview of the modification procedure

2. STYLE MODIFICATION PROCEDURE

A simple monophonic, symbolic music representation can be seen as a series of I note events N, where each note event n_i is a tuple of pitch p_i as MIDI note number and duration d_i in quarter lengths (ql).

$$N = (n_1, n_2, \dots n_I) \text{ where } n_i = (p_i, d_i) \qquad (1)$$

In the case of rest, a predefined rest symbol takes the place of note number. Along with the note events a chord annotation is needed, also being represented as a series of J chord events C, where each chord event c_j is a tuple of chord symbol s_j and duration d_j, again in ql.

$$C = (c_1, c_2, \dots c_J) \text{ where } c_j = (s_j, d_j) \qquad (2)$$

[1] Translation by the author. Original version:
So muß man sich mit Umschreibungen begnügen. Der Stil ist das Zentrum künstlerischer Behandlung und Erfassung, er erweist sich, wie Goethe sagt, als eine Erkenntnisquelle von viel tieferer Lebenswahrheit, als die bloße sinnliche Beobachtung und Nachbildung.

Such an existing N with a corresponding C should be modified in such a way that it comes closer to a specific style. C is fixed and won't be changed. This task is viewed as a local search, performing a multi-objective optimization for finding a local-optimal N [3]. The main idea is to start from a piece of music and try out *neighbors*. If a neighbor is better than the original one, according to the objectives described in section 4, the neighbor is saved and the process is iteratively continued with this new one. In a multi-objective optimization it is not straightforward what is considered *better*. For ensuring that no objective becomes worse, Pareto optimality is utilized: A state is only considered better, if all objectives stay the same or increase in value. A neighbor is reached by randomly doing one of the following changes:

- Changing the pitch p_i of a single note event. The maximum change interval is a major third upwards or downwards.
- Changing the duration of two notes d_{i_1} and d_{i_2} so that the overall duration of the note succession in this voice stays the same.
- A note n_i is divided into multiple ones, having $(1/2, 1/2)$, $(2/3, 1/3)$ or $(1/3, 1/3, 1/3)$ of the original duration d_i.
- Two successive notes n_i and n_{i+1} are joined into a single one with duration $d_i + d_{i+1}$.

Ideally one wants to have a manageable amount of neighbors so that every one can be tried to find out which one is best. Unfortunately the amount is not manageable in this case: For a given state with N note events, there are $16N^2 - 4N - 1$ possible neighbors (if one assumes that each possibility of splitting/joining notes is always valid). Because one cannot try out all possibilities, one randomly tries out neighbors and applies a technique from the toolbox of metaheuristics: Tabu search, which means storing the already tried out possibilities for not visiting them again. But even by this only a small fraction of possibilities will be tried out and one will be trapped in a local optimum. In most cases many efforts are employed to overcome local optima for finding the global one or at least solutions better than the first local optimum. But here there is a relaxing property: There is no need for finding the global optimum anyway. Changing a given piece of music for pushing it into a specific style should not mean to completely throw away the original piece and replace it by the stylistic *best* piece ever possible. It is reasonable doing small modifications until no modification improves the values of all objectives. By that it can be assured that the modified piece retains characteristics of the original one.

3. DATA CORPUS

Before formulating objectives, a data corpus of examples of music in the target style as well as counterexamples has to be established. The choice of counterexamples is crucial: Ideally a musical corpus would be needed that covers all music ever possible that doesn't belong to the target style in a statistically significant way. In practice this isn't possible. As an compromise, on the one hand it should be a style quite different from the target style, on the other hand it should be not too far off, so that the target style is sharpened by differentiating it from the counterexamples.

In our use case pieces of Jaco Pastorius (8 pieces of 2227.5 ql duration in total) form the target corpus whereas pieces of Victor Wooten (8 pieces of 3642.75 ql duration in total) form the counterexamples. So both corpora contain electric bass guitar music in the genre of jazz rock, whereas both bassists clearly show a different style. The complete list of pieces used is to be found in [4].

In the following section the objectives used in the Pastorius-project are introduced. But a key feature of this method is its flexibility: If some of its objectives don't seem appropriate or other objectives seem to bee needed for a specific target style, it is easy to leave some out and/or develop new ones.

4. OBJECTIVES

4.1 Feature Classification

Before tackling the modification task, a classification task is to be solved. For that purpose one has to design individual feature extractors, tailored to the specifics of the target style. If, like in our case, it is to be assumed that style is not only recognizable when the full piece has been played, but also on a local level, one can apply windowing of a fixed musical duration.

The feature extractors compute for each window a row vector x_i, forming the feature matrix X. A target value y_i can be assigned to each feature vector, numerically representing the target style with 0 and the style of the counterexamples with 1. So for the corpus the target values are obvious, forming the column vector Y. Therefore a function $f : X \rightarrow Y$ is needed. This can be learned from the data corpus, where cross validation assures a certain generalizability. For learning this function we made good experiences with Gradient Tree Boosting, because of its good performance as well as the interpretability of the utilized decision trees. For details regarding Decision Trees and Gradient Tree Boosting see [5]. The Gradient Tree boosting classifier cannot only classify a given window of music, it also can report a probability of a the window belonging to the target style. This probability forms the first objective that is to be optimized.

This objective is the most flexible one because the feature design can be customized for the characteristics that the modified music should have. Adopting this method for new styles goes hand in hand with a considerable work in designing appropriate feature extractors.

Beside 324 feature dimensions coming from already implemented feature extractors of music21 [6], multiple feature extractors have been customly designed for the music of the Pastorius project, outputting 86 dimensions. As an example a feature extractor should be shortly described that aims to model one of the striking features of the bass guitar play of Jaco Pastorius, according Pastorius-expert Sean Malone:

> Measure 47 [of *Donna Lee*, author's note] contains the first occurrence of what would become a Pastorius trademark: eighth-note triplets in four-note groups, outlining descending seventh-chord arpeggios. The effect is polyrhythmic – the feeling of two separate pulses within the bar that don't share an equal division. [...] As we will see, Jaco utilizes this same technique (including groupings of five) in many of his solos. [7, p. 6]

For calculating this feature f_{Jaco}, firstly the lengths of the sequences of notes with common direction are determined. Common direction means either successively ascending or descending in pitch. The length occurring most often is called the most common sequence length f_{Len}. So this feature is calculated by taking the fractional part of the quotient of the most common sequence length and the denominator of most common quarter length duration f_{Dur}.

$$f_{\text{Jaco}} = \frac{f_{\text{Len}}}{b} \bmod 1 \text{ , where } \frac{a}{b} = f_{\text{Dur}}, \quad \gcd(a,b) = 1 \qquad (2)^{2}$$

See figure 2 for an example, where $f_{\text{Len}} = 4$ and $f_{\text{Dur}} = 1/3$, so $f_{\text{Jaco}} = 4/3 \bmod 1 = 1/3$.

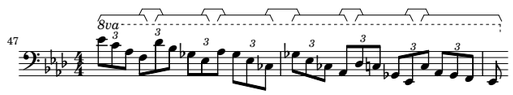


Figure 2: Bar 47–48 of Pastorius' *Donna Lee*. Brackets indicate sequences of notes with common direction. [3]

The complete list and description of the features extractors used is to be found in [4].

4.2 Markov Classification

While the classification described in the previous section captures general characteristics of the music, depending on the feature extractors, Markov chains [8] are an old friend for music generation that works well on a local level. The first-order Markov model assumes a fixed set of possible note events $S = \{s_1, s_2, \ldots, s_K\}$ and assigns a probability a_{kl} for each note event s_k being preceded by a note event s_l.

$$a_{kl} \equiv P(n_i = s_k | n_{i-1} = s_l) \text{ with } a_{kl} \geq 0 \text{ and } \sum_{l=1}^{K} a_{kl} = 1 \qquad (3)$$

Along with the $K \times K$ transition matrix a_{kl} the initial probabilities π_k for note events without predecessor are needed.

$$\pi_k \equiv P(n_1 = s_k) \text{ with } \pi_k \geq 0 \text{ and } \sum_{k=1}^{K} \pi_k = 1 \qquad (4)$$

The assumption of the dependency of a fixed number of O predecessors is called the order of the Markov chain. First order chains, like in equations 3–5, clearly are an unrealistic model, but for music, even when $\lim_{O\to\infty}$ an Oth-order Markov chain wouldn't hold true, because the probability of a note can even be influenced by its successors. [4] Nevertheless a sufficiently large order O usually captures good local characteristics.

For optimizing a given note sequence N of length I, the mean probability is used as objective.

$$\bar{P}(N|a_{kl},\pi_k) = \frac{P(n_1)\cdot\prod_{i=2}^{I} P(n_i|n_{i-1})}{I} \qquad (5)$$

[2] gcd means greatest common divisor. This line is just for the purpose to indicate that a/b is irreducible in lowest terms.

[3] This and the following Pastorius music examples in this paper are newly typeset with [7] as reference.

[4] Think of the climax of a musical phrase that is headed for already some time before.

Some further adjustments have been made to improve the results:

- Separate chains have been trained for durations and pitches for getting less sparse probability matrices.
- Since we assume chord annotations, separate chains can be trained for each chord symbol type.
- Linear interpolation smoothing [5] as well as additive smoothing [6] has been applied. Both smoothing techniques counteract the zero-frequency problem, i.e. the problem of yet unseen data. The first one means to take the average of several order Markov chains ($0 \leq O \leq 4$ in our project) and the latter one to add a tiny constant term to all probabilities.

See figure 3 for a graphical depiction of this objective function based on real data of Jaco Pastorius. The optimization can be imagined as an hill climbing on this surface plot.

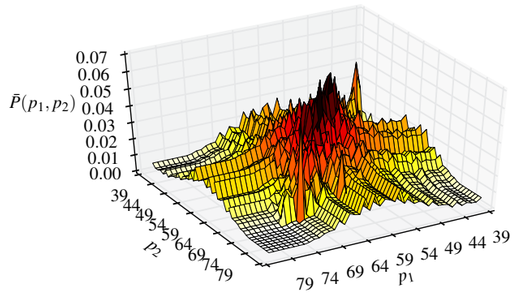


Figure 3: Surface of an objective function of two successive pitches regarding the average smoothed Markov probability (orders 0 and 1) for the Pastorius-project

4.3 Ratio of example/counter-example Markov probability

The preceding objective only takes the Markov model of the target style into account. So it also rewards changes that make a given note succession more close to general musical characteristics. To foster the specific characteristics of the target style, the value of this objective is the ratio of the average smoothed Markov probability of the target style and average smoothed Markov probability of counterexamples. [7]

Figure 4 shows a graphical depiction of this objective function. Compare with figure 3 to clarify the big difference between this and the previous objective.

[5] See [9] for a general depiction and [10] for one referring to Markov model music generation.

[6] See [11] for a comparison of different smoothing techniques. There on p. 311 it is argued that additive smoothing generally performs poorly, but note in the case of this project it is used in combination with interpolation smoothing (related to what is there called Jelinek-Mercer-Smoothing).

[7] A possibility for improving this objective is applying Bayes' theorem. This objective could than be reformulated (with target being the target style and counter being the style of the counterexamples):

$$P(\text{target}|N) = \frac{P(N|\text{target})P(\text{target})}{P(N)} \text{ and}$$
$$P(N) = P(N|\text{target})P(\text{target}) + P(N|\text{counter})P(\text{counter})$$

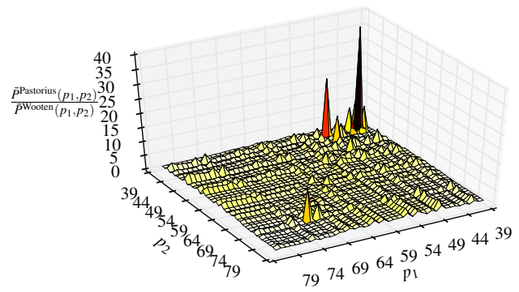


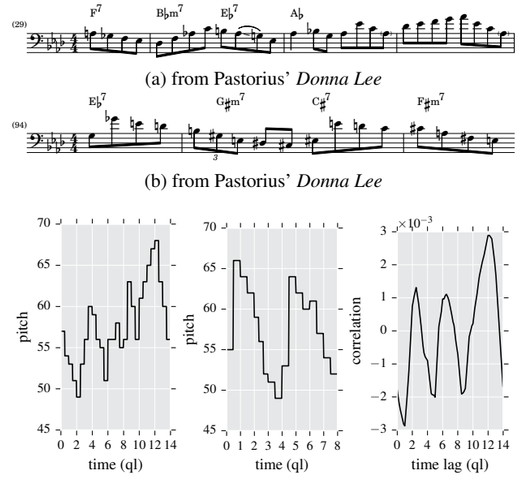
Figure 4: Surface of an objective function of two successive pitches regarding the ratio of the example/counter-example-ratio for the average smoothed Markov probabilities (order 0 and 1) with Jaco Pastorius being the target style and Victor Wooten being the counter-example.

4.4 Time correlations for chord-repetitions

The main idea of this objective is to capture some large scale structural similarities within the pieces of the target style. A shallow idea of large scale structure should also be given by the feature classification (subsection 4.1) when the relative position of the window is given as feature. But in practice large scale structure is something one misses most in the generated music. So this is an additional approach to include that. Our approach is based on two assumptions about repetitions in music: The first assumption is, that structure evolves by the absence or presence of repetition, e.g. the varied reoccurrence of material already played some time ago. The second assumption is that such kind of repetition most probably occurs when the relative changes in harmony also repeat. For each such a repetition both corresponding subparts of the note sequence N is converted into a piano roll representation (cf. figure 5c and d) where, after subtracting the mean and normalizing, a circular cross correlation (cf. figure 5e) is performed, so this objective correlates with motivic similarity. When referring to both parts as $N^{(1)}$ and $N^{(2)}$ with length I, the circular cross correlation is given by:

$$R_{N^{(1)}N^{(2)}}(l) = \frac{1}{I}\sum_{i=1}^{I} N_i^{(1)} N_{(i+l) \bmod I}^{(2)} \qquad (6)$$

And that's how it is applied in the optimization: All cross-correlations between parts with related chord-progressions in the target style corpus are pre-computed. During the optimization the cross-correlations of those related sections are computed, too. Then the inner product between each correlation of the piece of music to be optimized and the ones of the target style corpus are computed and the maximum one is returned as value of this objective. Since the dot product of correlations of different lengths cannot be computed, all correlations of the same progression length are brought to the same length by linear interpolation. By that means one ensures that the correlation within the chord progressions in the music to be optimized becomes more similar to ones of the examples in the target style.



(a) from Pastorius' *Donna Lee*

(b) from Pastorius' *Donna Lee*

(c) Piano roll representation of a.

(d) Piano roll representation of b.

(e) Cross-correlation of c and d.

(f) Aligning of the two examples with b being circularly shifted to the maximum value of e.

Figure 5: Illustration of this objective for two examples with the same intervals between the roots of the chord progression (when enharmonic change is ignored). The cross-correlation shows that a circular shifting of 12 ql the two examples have the maximal motivic similarity.

5. RESULTS

In contrast to the classification, where the performance can be evaluated relatively impartially, such an objective evaluation is not possible for the modification. One can only try it out and evaluate the result subjectively. So the assessment heavily depends on the judges, their musical and cultural background, taste. As an example, a simple traditional melody has been chosen: *New Britain*, often sung along with the Christian hymn *Amazing Grace*. See figure 6 for the original version. During the modification process, 3398 different changes have been tried out whereby only 14 ones have been accepted by Pareto optimality. See figure 7 for the final version.

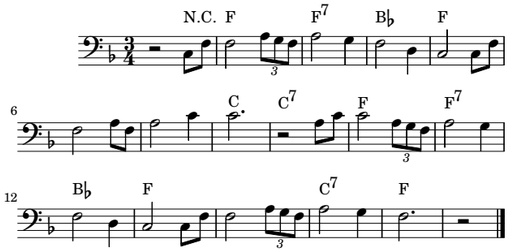


Figure 6: *New Britain* resp. *Amazing Grace*, original version.

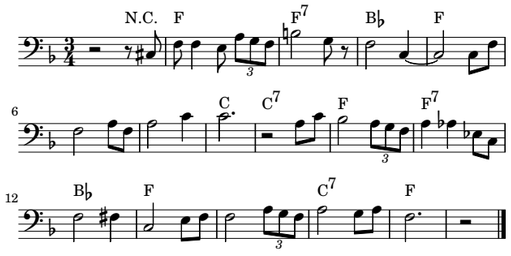
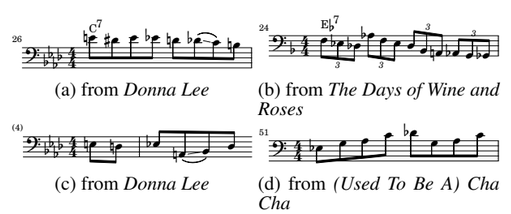


Figure 7: *New Britain* resp. *Amazing Grace*, modified version.

The tieing of notes from bars 4 to 5 seems to interrupt the arc of suspense, thus it seems to be rather unfavourable from a musical viewpoint. The run from bar 9 seems interesting and coherent. The two eighths in bar 9 as well as the ones in bar 15 seem to surround the following principal note, which lets the music appear quite natural. Especially the first example can be considered a broadened *double appoggiatura*. Some successions seem harsh, like the succession of the major to the minor third of the chord in bar 11 or the melodic succession of a semitone and a tritone in bars 12–13, but such harsh elements are not unusual in the music of Pastorius, see figure 8. Maybe they seem spurious, since one rather bears in mind the consonant original version, which is still noticeable in the modified version to a large extent. But from that point of view, the result is a successful blending of the original version and the style of Pastorius, even if it is doubtful if Pastorius would have improvised over *New Britain* like this.



(a) from *Donna Lee* (b) from *The Days of Wine and Roses*

(c) from *Donna Lee* (d) from *(Used To Be A) Cha Cha*

Figure 8: Jaco Pastorius examples, that could be the model for *harsh* results in the modification. a and b: Succession of the minor on the major third of the chord. c and d: Melodic succession of a semitone and a tritone.

6. RELATED WORK

One of the most prominent researchers, engaging computationally with musical style, especially in symbolic style synthesis, is David Cope [12–14]. In its basic form his style replication program EMI ("Experiments in Musical Intelligence") has to be fed by over a thousand of user input questions. Cope also attempts to overcome this by automatically analyzing a corpus of music. Roughly, this involves finding what Cope calls signatures, frequently reoccurring sequences, assigning functional units to them and recombining the corpus with special regards to those functional signatures. This leads to impressive results for music with rather homogeneous texture, but it may be less appropriate for more eclectic and erratic styles, like the one of Pastorius, because signatures are less dominant. Nevertheless, it would be interesting to apply Copes inspiring work to eclectic and erratic styles since Cope developed his methodology sophisticatedly, far exceeding the rough and basic ideas just touched here.

Cope describes basic categories into which music composing programs fall:

> The approaches [...] include rules-based algorithms, data-driven-programming, genetic algorithms, neural networks, fuzzy logic, mathematical modeling, and sonification. Although there are other ways to program computers to compose music, these seven basic processes represent the most commonly used types. [12, p. 57]

If one would like to force the approach of this paper to fall into these categories, rules-based programming [8] and data-driven-programming would fit, but a considerable amount of this work wouldn't be described. Especially Cope's category of *genetic algorithms* (GAs) is too specific and could be generalized to *metaheuristics*, which then would also fit for the method described here. GAs enable random jumps in the optimization neighborhood whereas the method presented here only takes small steps for ensuring that a local optimum is targeted—which is desired as described in section 2. Markov chains have a great tradition in music. They found application very early in both computer aided music generation [15–17] and in musicological studies [18, 19]. More recently, researchers from the Sony Computer Science Laboratory rediscovered Markov chains by combining them with constraint based programming, yielding very interesting results [10, 20]. In general, most music generation approaches, including Cope's and all Markovian methods, are united by the strategy of recombining elements of an existing musical corpus. Other attempts that fall under this umbrella are suffix based methods [21, 22]. The method presented here, however, also enables recombinatorial results, but is not restricted to that because the other optimization objectives also foster the generation of music that is similar to the corpus on a more abstract level. [22, 23] share the concept of an underlying harmonic progression with the approach presented here. [24] is similar in the approach to apply metaheuristics for musical style modification, but is not about learning the objectives from a data corpus in the way described here.

Having mentioned some major branches of automatic music generation, the author recommends a more complete survey [25] for those interested in more branches of this field.

7. CONCLUSIONS

In this paper a novel approach of musical style modification has been presented. Basically this is a multi-objective optimization, where the objectives try to reward similarity to the target style in different respects. By that means a given piece of music can be transformed with the aim of pushing it closer to a specified target style. There are plenty of possibilities to built upon this work: making it real-time capable (currently it is not), paying more regard to the metrical structure (a weakness in the Pastorius-project), validly evaluating the modification results by empirical experiments. We also started to conceptualize about pedagogical applications: during the modification process, *reasons* why things has been changed in certain manner, can be tracked—something that could be expanded for automatically explaining style. This shows that the potential of this approach is far from exhausted.

[8] That fits since in Cope's terminology Markovian processes fall into this category

Acknowledgments

This paper is a condensed form of a Master thesis [4], so thanks are owed to all who supported this thesis: First and foremost SAP SE who generously supported it in financial terms, by a providing a great working environment as well as with personnel support: Stephan Brand who initiated the project and debated about the project from the perspective of a bassist, Pastorius enthusiast and software manager; Benjamin Graf who was available weekly for discussions especially about data science issues. Special thanks to Prof. Denis Lorrain whose encouraging and far-sighted arguments have been inspiring as well as Sean Malone, who contributed by occasional mail contact with beneficial comments from his view as bassist, musicologist and Pastorius expert.

8. REFERENCES

[1] G. Adler, *Der Stil in der Musik. 1. Buch: Prinzipien und Arten des Musikalischen Stils*. Leipzig: Breitkopf & Härtel, 1911.

[2] L. B. Meyer, "Meaning in Music and Information theory," in *The Journal of Aesthetics and Art Criticism*, vol. 15, no. 4, 1957, pp. 412–424.

[3] S. Luke, *Essentials of Metaheuristics*, 2nd ed. Raleigh: Lulu, 2013. [Online]. Available: https://cs.gmu.edu/~sean/book/metaheuristics/

[4] F. Zalkow, "Automated musical style analysis. Computational exploration of the bass guitar play of Jaco Pastorius on symbolic level," Master's Thesis, University of Music Karlsruhe, Germany, Sep. 2015.

[5] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, *The elements of statistical learning. Data mining, inference, and prediction*, ser. Springer series in statistics. New York: Springer, 2009.

[6] M. S. Cuthbert, C. Ariza, and L. Friedland, "Feature Extraction and Machine Learning on Symbolic Music using the music21 Toolkit," in *Proceedings of the 12th International Symposium on Music Information Retrieval*, 2011, pp. 387–392.

[7] S. Malone, *A Portrait of Jaco. The Solo Collection*. Milwaukee: Hal Leonard, 2002.

[8] E. Alpaydın, *Introduction to Machine Learning*, 2nd ed., ser. Adaptive computation and machine learning. Cambridge and London: MIT Press, 2010.

[9] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Proceedings, Workshop on Pattern Recognition in Practice*. Amsterdam: North Holland, 1980, pp. 381–397.

[10] F. Pachet and P. Roy, "Markov constraints. Steerable generation of Markov sequences," in *Constraints*, vol. 16, no. 2, March 2011, pp. 148–172.

[11] S. F. Chen and J. Goodman, in *An empirical study of smoothing techniques for language modeling*, vol. 13, no. 4, 1999, pp. 359–393.

[12] D. Cope, *Computers and Musical Style*. Madison: A-R Editions, Inc., 1991.

[13] ——, *Virtual Music. Computer Synthesis of Musical Style*. Cambridge: MIT Press, 2001.

[14] ——, *Computer Models of Musical Creativity*. Cambridge: MIT Press, 2005.

[15] F. Brooks, A. Hopkins, P. Neumann, and W. Wright, "An experiment in musical composition," in *IRE Transactions on Electronic Computers*, vol. 6, no. 3, 1957, pp. 175–182.

[16] L. Hiller and L. Isaacson, *Experimental Music. Composition With an Electronic Computer*. New York: McGraw-Hill, 1959.

[17] I. Xenakis, *Formalized Music. Thought and Mathematics in Composition*, 2nd ed., S. Kanach, Ed. Stuyvesant: Pendragon Press, 1992.

[18] R. C. Pinkerton, "Information theory and melody," in *Scientific American*, vol. 194, no. 2, 1956, pp. 77–86.

[19] J. E. Youngblood, "Style as Information," in *Journal of Music Theory*, vol. 2, no. 1, 1958, pp. 24–35.

[20] F. Pachet, P. Roy, and G. Barbieri, "Finite-length Markov Processes with Constraints," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, vol. 1, 2011, pp. 635–642.

[21] S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano, "Using Machine-Learning Methods for Musical Style Modeling," *Computer*, vol. 36, no. 10, pp. 73–80, Oct. 2003.

[22] J. Nika and M. Chemillier, "Improtek: integrating harmonic controls into improvisation in the filiation of OMax," in *Proceedings of the 2012 International Computer Music Conference*, 2012, pp. 180–187.

[23] A. Donzé, R. Valle, I. Akkaya, S. Libkind, S. A. Seshia, and D. Wessel, "Machine Improvisation with Formal Specifications," in *Proceedings of the 40th International Computer Music Conference*, 2014, pp. 1277–1284.

[24] T. Dimitrios and M. Elen, "A GA Tool for Computer Assisted Music Composition," in *Proceedings of the 2007 International Computer Music Conference*, 2007, pp. 85–88.

[25] J. D. Fernández and F. Vico, "AI Methods in Algorithmic Composition: A Comprehensive Survey," in *Journal of Artificial Intelligence Research*, vol. 48, no. 1, 2013, pp. 513–582.

Synchronization of van der Pol Oscillators with Delayed Coupling

Andreas Henrici
Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering
Technikumstrasse 9
CH-8401 Winterthur, Switzerland
henr@zhaw.ch

Martin Neukom
Zürcher Hochschule der Künste
Institute for Computer Music and Sound Technology
Toni-Areal, Pfingstweidstrasse 96
CH-8031 Zürich, Switzerland
martin.neukom@zhdk.ch

ABSTRACT

The synchronization of self-sustained oscillators such as the van der Pol oscillator is a model for the adjustment of rhythms of oscillating objects due to their weak interaction and has wide applications in natural and technical processes. That these oscillators adjust their frequency or phase to an external forcing or mutually between several oscillators is a phenomenon which can be used in sound synthesis for various purposes. In this paper we focus on the influence of delays on the synchronization properties of these oscillators. As there is no general theory yet on this topic, we mainly present simulation results, together with some background on the non-delayed case. Finally, the theory is also applied in Neukom's studies 21.1-21.9.

1. INTRODUCTION

If several distinct natural or technical systems interact with each other, there is a tendency that these systems adjust to each other in some sense, i.e. that they synchronize their behavior. Put more precisely, by synchronization we mean the *adjustment of the rhythms of oscillating objects due to their mutual interaction*. Synchronization can occur in model systems such as a chain of coupled van der Pol oscillators but also in more complex physical, biological or social systems such as the coordination of clapping of an audience. Historically, synchronization was first described by Huygens (1629-1695) on pendulum clocks. In modern times, major advances were made by van der Pol and Appleton. Physically, we basically distinguish between synchronization by external excitation, mutual synchronization of two interacting systems and synchronization phenomena in chains or topologically more complex networks of oscillating objects. In this paper, we will focus on the case of either two interacting systems or a chain of a small number of oscillators.

Clearly, the synchronizability of such a system of coupled oscillators depends on the strength of the coupling between the two oscillators and the detuning, i.e. the frequency mismatch of the two systems. If the coupling between the two systems does not happen instantaneously, but with a delay, the question of the synchronizability becomes much more difficult to answer. The assumption of delayed feedback however is a vary natural one, since most natural and technical systems do not answer instantaneously to external inputs, but rather with a certain delay, due to physical, biological, or other kinds of limitations. The effect of using delays can be easily modeled in sound synthesis applications and therefore allows a fruitful exchange between theoretical and empirical results on the one hand and musical applications on the other hand.

In the absence of synchronization, other effects such as beats or amplitude death become important, and these effects depend (besides the coupling strength and the frequency detuning) on the delay of the coupling between the oscillators as well.

Self-sustained oscillators can be used in sound synthesis to produce interesting sounds and sound evolutions in different time scales. A single van der Pol oscillator, depending on only one parameter (μ, see (1)), produces a more or less rich spectrum, two coupled oscillators can synchronize after a while or produce beats depending on their frequency mismatch and strength of coupling [1,2]. In chains or networks of coupled oscillators in addition different regions can synchronize (so-called chimeras), which takes even more time. If the coupling is not immediate but after a delay it can take a long time for the whole system to come to a steady or periodic changing state. In addition all these effects can not only be used to produce sound but also to generate mutually dependent parameters of any sound synthesis technique.

This paper gives an introduction to the theory of synchronization [3,4] and shows how to get discrete systems, that is difference equations, from the differential equations and shows their usage in electroacoustic studies of one of the authors (Neukom, Studien 21.1 - 21.9.).

2. SYNCHRONIZATION OF COUPLED OSCILLATORS

2.1 Self-sustained oscillators

Self-sustained oscillators are a model of natural or technical oscillating objects which are active systems, i.e. which contain an inner energy source. The form of oscillation does not depend on external inputs; mathematically, this corresponds to the system being described by an autonomous (i.e. not explicitly time-dependent) dynamical system. Under perturbations, such an oscillator typically returns to the original amplitude, but a phase shift can remain even under weak external forces. Typical examples of self-sustained oscillators are the van der Pol oscillator

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -\omega_0^2 x + \mu(1+\gamma-x^2)y \end{aligned} \tag{1}$$

or the Rössler or Lorenz oscillators. Note that in the van der Pol oscillator (1), the parameters μ and γ measure the strength of the nonlinearity; in particular, for $\mu = 0$ we obtain the standard harmonic oscillator. In the case of a single oscillators, we usually set $\gamma = 0$, in the case of several oscillators however, we can use distinct values of γ to describe the amplitude mismatch of the various oscillators. Assuming $\gamma = 0$, in the nonlinear case $\mu \neq 0$, the term $\mu(1-x^2)y$ means that for $|x| > 1$ and $|x| < 1$ there is negative or positive damping, respectively.

In the nonlinear case, these systems cannot be integrated analytically, and one has to use numerical algorithms (and also take into account the stiffness of e.g. the van der Pol system for large values of μ). One can also consider discrete systems of the type

$$\phi^{(k+1)} = F(\phi^{(k)}), \tag{2}$$

which often occurs in cases when one can measure a given systems only at given times $t_0, t_1, \ldots$.

We will discuss an implementation of the van der Pol model (1) in section 5.

2.2 Synchronization by external excitation

The question of synchronization arises when systems of the type (1) and (2) are externally forced or connected together. As a generalization of the van der Pol system (1), weakly nonlinear periodically forced systems are of the form

$$\dot{x} = F(x) + \epsilon p(t), \tag{3}$$

where the unforced system $\dot{x} = F(x)$ has a stable T_0-periodic limit cycle $x_0(t)$ and $p(t)$ is a T-periodic external force. The behavior of the system then primarily depends on the amplitude ϵ of the forcing and the frequency mismatch or detuning $\nu = \omega - \omega_0$, where ω_0 and ω are the frequencies of the oscillator (1) and the T-periodic external force $p(t)$, $\omega = \frac{2\pi}{T}$. One can show that in the simplest case of a sinusoidal forcing function the dynamics of the perturbed system (3) can be described by the *Adler equation*

$$\dot{\theta} = -\Delta + \epsilon \sin(\theta) \tag{4}$$

for the relative or slow phase $\theta = \phi - \omega t$. A stable steady state solution of (4) exists in the case

$$|\Delta| < |\epsilon| \tag{5}$$

and corresponds to a constant phase shift between the phases of the oscillator and the external forcing. The condition (5) describes the synchronization region in the Δ-ϵ-parameter space. Outside the synchronization region, one observes a beating regime with beat frequency

$$\Omega = 2\pi \left(\int_0^{2\pi} \frac{\mathrm{d}\psi}{\sqrt{\epsilon \sin(\psi) - \nu}} \right)^{-1}. \tag{6}$$

2.3 Mutual synchronization and chains of oscillators

Here we consider two coupled systems of the type (3), namely

$$\begin{aligned} \dot{x}_1 &= F_1(x_1, x_2) + \epsilon p_1(x_1, x_2), \\ \dot{x}_2 &= F_2(x_1, x_2) + \epsilon p_2(x_1, x_2) \end{aligned} \tag{7}$$

In the case of weak coupling, i.e. $\epsilon \ll 1$, (7) can be reduced to an equation for the phase difference $\psi = \phi_1 - \phi_2$ of the type (4), and the synchronization region is again of the type (5), where Δ in this case is the difference between the frequencies of the unperturbed oscillators x_1 and x_2. If the coupling becomes larger, the amplitudes have to be considered as well.

To be specific, we consider two coupled van der Pol oscillators, which we assume to connected by a purely dissipative coupling, which is measured by the parameter β:

$$\begin{aligned} \ddot{x}_1 + \omega_1^2 x_1 &= \mu(1-x_1^2)\dot{x}_1 + \mu\beta(\dot{x}_2 - \dot{x}_1), \\ \ddot{x}_2 + \omega_2^2 x_2 &= \mu(1+\gamma-x_2^2)\dot{x}_2 + \mu\beta(\dot{x}_1 - \dot{x}_2). \end{aligned} \tag{8}$$

Here the two oscillators have the same nonlinearity parameter μ, and γ and $\Delta = \omega_2 - \omega_1$ describe the amplitude and frequency mismatches. In Figure 1, we show the results of a numerical computation of the synchronization region (which is usually called "Arnold tongue") of the system (8) in the case $\gamma = 0$ (no amplitude mismatch).

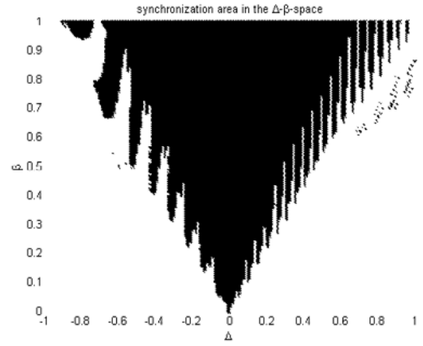


Figure 1. Synchronization area for two coupled van der Pol oscillators

If one considers an entire chain of oscillators (instead of $N = 2$ ones as in (8)), the model equations are for any $1 \leq j \leq N$

$$\ddot{x}_j + \omega_j^2 x_j = 2\epsilon(p - x_j^2)\dot{x}_j + 2\epsilon d(\dot{x}_{j-1} - 2\dot{x}_j + \dot{x}_{j+1}) \tag{9}$$

together with the (free end) boundary conditions $x_0(t) \equiv x_1(t)$, $x_{N+1}(t) \equiv x_N(t)$; sometimes we also use periodic boundary conditions, i.e. $x_0(t) \equiv x_N(t)$, $x_1(t) \equiv x_{N+1}(t)$. On the synchronization properties of chains as given by (9), in particular the dependence on the various coupling strengths (which can also vary instead of being constant as in (9)), there exists a vast literature, we only mention the study [5].

In this paper we restrict our attention to the model (8) of $N = 2$ oscillators; in our musical application however, we consider chains of the type (9) for $N = 8$. Our goal is to study how the Arnold tongue (Figure 1) is deformed when delays are introduced onto the model.

3. INFLUENCE OF DELAYS ON SYNCHRONIZATION

3.1 Arnold tongue of synchronization

If the coupling between the oscillators occurs with certain delays, we obtain instead of (8) the following model, again considering only dissipative coupling:

$$\begin{aligned}\ddot{x}_1(t) + \omega_1^2 x_1(t) &= \mu(1 - x_1(t)^2)\dot{x}_1(t) \\ &\quad + \mu\beta_{21}(\dot{x}_2(t-\tau_A) - \dot{x}_1(t-\tau_1)), \\ \ddot{x}_2(t) + \omega_2^2 x_2(t) &= \mu(1 + \gamma - x_2(t)^2)\dot{x}_2(t) \\ &\quad + \mu\beta_{12}(\dot{x}_1(t-\tau_B) - \dot{x}_2(t-\tau_1))\end{aligned} \tag{10}$$

Here we have 3 different delays, namely τ_1, τ_A and τ_B, which are the delays of self-connection, from oscillator x_2 to x_1 and from x_1 to x_2, respectively. Similarly, we have 2 different feedback factors, namely β_{21} and β_{12}, which describe the feedback strength from oscillator x_2 to x_1 and from x_1 to x_2, respectively.

To investigate the influence of the delays on the synchronization of the oscillators, we simulated the system (10) numerically again for $\gamma = 0$ (no amplitude mismatch) and for identical delays $\tau := \tau_1 = \tau_A = \tau_B$ ranging from $\tau = 0$ (no delay) to $\tau = 2$. In Figure 2 we show the Arnold tongue of the system for various values of τ in the described interval. Here we set $\beta := \beta_{12} = \beta_{21}$; note however that in section 4 we will also consider the case $\beta_{12} \neq \beta_{21}$.

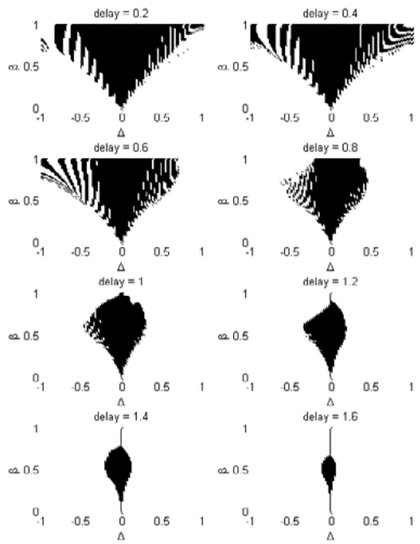


Figure 2. Synchronization area for two coupled van der Pol oscillators for various delay values

An analytical investigation of the synchronization area for growing values of τ is beyond the scope of this paper, but can be accomplished based on the analysis of the non-delayed case (see [3]) with additionally using methods for dealing with time-delay systems (see [6]). For a study of a single van der Pol oscillator with delayed self-feedback see [7].

3.2 Dependence of the beat frequency on the delay

As explained in section 2.2, outside the synchronization region, the dynamics of the system of coupled oscillators can be described by the beating frequency, namely the frequency of the relative phase of the two oscillators. In the case of an externally forced single oscillator, the beating frequency is given by the formula (6).

For large values of the coupling, besides the synchronization and beating regimes, one also observes the phenomenen of *oscillation death*. More precisely, oscillation death occures when the zero solution of the equations (10) becomes stable, which in the absence of delay it only is for large values of the coupling β. For some results on the dependence of the amplitude death region on the delay parameter τ in the case without detuning, we refer to [8]. We do not discuss this topic in detail, since it is of minor interest for our applications.

Of great relevance for our application however is the understanding of the beating regime, in particular the dependence of the beating frequency on the delay parameter τ. While an analytic discussion of the influence of the parameters τ (delay), Δ (detuning), β (coupling strength) and μ (nonlinearity) is beyond the scope of this paper, we present some results of numerical simulations. In this section we focus on investigating the combined influence of Δ and τ on the beat frequency Ω, i.e. the behavior of the beat frequency in the delay-detuning space, while in the following section, which is devoted to an implementation of the system of coupled van der Pol oscillators in Max, we focus on the behavior of Ω in the delay-coupling-space.

The following figures (Figures 3 and 4) show the beat frequency in the τ-Δ-space for $\mu = 1$ and $\beta = 0.5$; darker colors signify a higher beat frequency, i.e. the white region of the space belongs to the synchronization region.

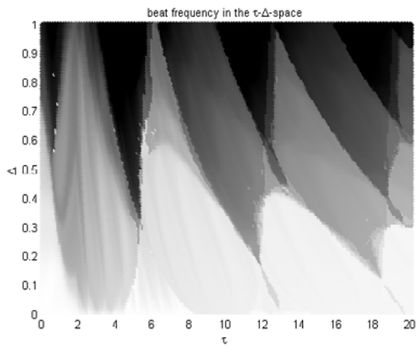


Figure 3. The beat frequency as a function of τ and Δ for $\mu = 1$ and $\beta = 0.5$

One can observe that for a given value of the delay τ, the beat frequency grows with the detuning, which is intuitively plausible, while for a given value of the detuning Δ, the beat frequency varies periodically with the delay τ, which is in good accordance with the results of the simulations in Max presented in section 4.

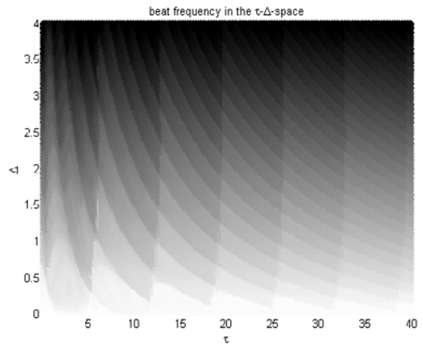


Figure 4. The beat frequency as a function of τ and Δ for $\mu = 1$ and $\beta = 0.5$

4. IMPLEMENTATION IN MAX

In order to experiment in real time we implemented the van der Pol oscillator in Max. While for the production of the figures in sections 2.3 and 3.1 we used the `ode45`- and `dde23`-method of `Matlab`, we will now show explicitly how to obtain discrete systems of the type (2), that is difference equations, from the differential equations: first by Euler's Method used in the studies 21 and then the classical Runge-Kutta method implemented in the Max-patch `icmc16_vdp.maxpat` which we used to produce the following figures. The examples are programmed as mxj~ externals. The following Java code samples are taken from the perform routine of these externals. The externals and Max patches can be downloaded from [9].

The implementation of Euler's Method is straightforward, the code is short and fast and with the sample period as time step quite precise [1]. First the acceleration is calculated according to the differential equation above (1). Then the velocity is incremented by the acceleration times dt and displacement x by velocity times dt ($dt = 1$).

```
a = (− c∗x + mu∗(1 − x∗x)∗v);
      // with c = (frequency∗2∗Pi/sr)^2
v += a;
x += (v + in[i]);
```

The classical Runge-Kutta method (often referred to as RK4) is a fourth-order method. The values x and v of the next sample are approximated in four steps. The following code sample from the mxj~ external `icmc_vdp` shows the calculation of the new values x and v using the function `f_` which calculates the acceleration.

```
double f_(double x, double v){return − c∗x + mu∗(1 −
    x∗x)∗v;}

k1 = f_(x, v);
l1 = v;
k2 = f_(x+l1/2, v+k1/2);
l2 = v+k1/2;
k3 = f_(x+l2/2, v+k2/2);
l3 = v+k2/2;
k4 = f_(x+l3, v+k3);
l4 = v+k3;
v += (k1 + 2∗k2 + 2∗k3 + k4)/6 + in[i];
x += (l1 + 2∗l2 + 2∗l3 + l4)/6;
```

The next code sample shows how a delayed mutual feedback of two oscillators is implemented. The velocities of the two oscillators (`v1` and `v2`) are stored in the circular buffers `bufv1` and `bufv2`. The differences of the delayed velocities are multiplied by the feedback factor `fbv21` and `fbv12` respectively and added to the new velocities.

```
v1 += ( (k11 + 2∗k21 + 2∗k31 + k41)/6 +
    fbv21∗(buf2v[pout2] − buf1v[pout1]) + in[i] );
x1 += (l11 + 2∗l21 + 2∗l31 + l41)/6;

v2 += ( (k12 + 2∗k22 + 2∗k32 + k42)/6 +
    fbv12∗(buf1v[pout1] − buf2v[pout2]) + in[i] );
x2 += (l12 + 2∗l22 + 2∗l32 + l42)/6;
```

In the Max-patch `icmc16_vdp.maxpat` the beats are measured and plotted in a lcd object (Figure 5) as a function of the delay (in samples).

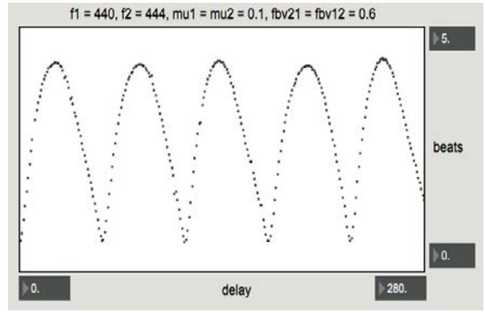


Figure 5. The beat frequency as a function of the delay

The following figures show the beat frequency as a function of the delay and the feedback in a 3D plot. More precisely, Figure 6 shows the results of the simulation of the Max-patch for the delay values 1,2, ...280 samples and the feedback fbv21 = fbv12 = 0, 0.1, ..., 0.7, and Figure 7 shows the analogous results for the delay values 1,2, ...160 samples and the feedback fbv21 = 0.4, fbv12 = 0, 0.2, ..., 3.0.

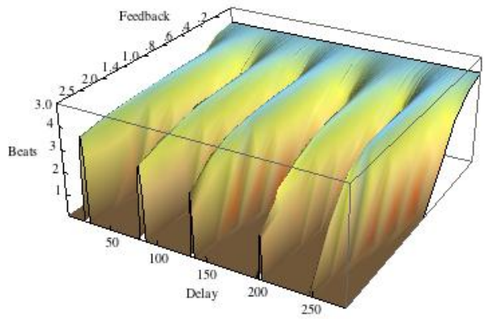


Figure 6. The beat frequency as a function of the delay and the feedback factor fbv21 = fbv12

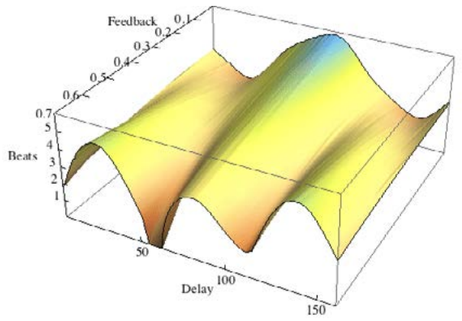


Figure 7. The beat frequency as a function of the delay and the feedback factor fbv21 (fbv12 constant)

Figure 8 shows the dependence of the beats on higher delay values.

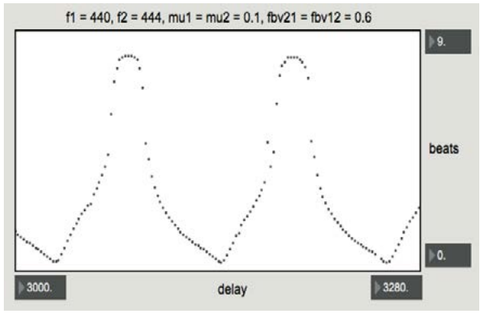


Figure 8. The beat frequency as a function of the delay for higher delay values

Especially in Figure 7, one can observe that an increase of the coupling generally leads to a decrease of the beating frequency, until it becomes zero, i.e. a transition to the synchronization region, with the exception of a periodic sequence of delay values with a higher beat frequency, which is however decreasing with increasing coupling as well. This transition towards synchronization can also be seen from the following sequence of plots (Figure 9):

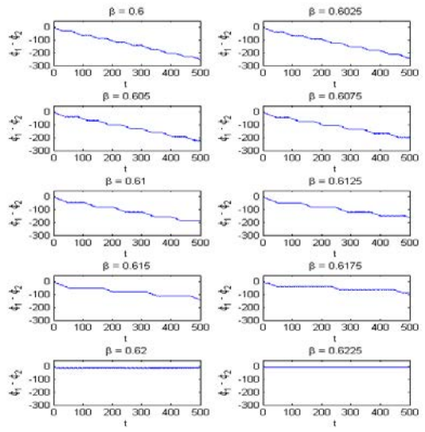


Figure 9. Transition from the beating to the synchronization regime

5. MUSICAL APPLICATIONS

In Neukoms 8-channel studies 21.1-21.9 eight van der Pol oscillators are arranged in a circle and produce the sound for the eight speakers, cf. equations (9) in the case of periodic boundary conditions. Each of these oscillators is coupled with its neighbors with variable delay times and gains in both directions. The main Max-patch contains eight joined sub-patches (Figure 10) which themselves contain the mxj~ external `m_vdp_del` and the delay lines (Figure 11).

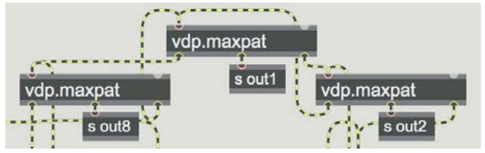


Figure 10. Three of the eight coupled sub-patches vdp.maxpat of the main Max-patch

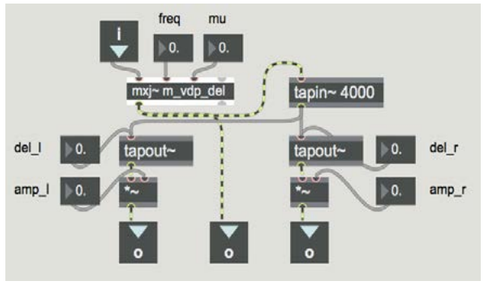


Figure 11. A simplified version of the vdp.maxpat showing the individual delays and gains to the left and to the right outlet and the direct output to the middle outlet

Two additional chains of eight van der Pol oscillators produce control functions which are used for amplitude and frequency modulation. If the frequencies of the oscillators are lower than about 20 Hz the modulations produce pulsations and vibratos. Depending on the coupling strength and the delay some or all pulsations and vibratos synchronize their frequencies. The relative phase which is not audible in audio range plays an important role in the sub-audio range: the pulsations of the single sound sources can have the same frequency while being asynchronous in a rhythmic sense. With growing coupling strength they can produce regular rhythmic patterns which are exactly in or out of phase.

The coupled van der Pol oscillators can be used as a system for purely algorithmic composition. Without changing any parameters the produced sound changes over a long time without exact repetitions. They also can be used as a stable system for improvisation with a wide range of sounds, rhythms and temporal behavior.

Some sound samples of a binaural version of Neukom's studies can be downloaded from [9].

6. REFERENCES

[1] M. Neukom, "Applications of synchronization in sound synthesis," in *Proceedings of the 8th Sound and Music Computing Conference SMC, 6. - 9. July 2011, Padova, Italy, 2011*.

[2] ——, *Signals, Systems and Sound Synthesis*. Peter Lang, 2013.

[3] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization*. Cambridge University Press, 2001.

[4] G. V. Osipov, J. Kurths, and C. Zhou, *Synchronization in Oscillatory Networks*. Springer-Verlag, 2007.

[5] T. V. Martins and R. Toral, "Synchronisation induced by repulsive interactions in a system of van der pol oscillators," *Progr. Theor. Phys.*, vol. 126, no. 3, pp. 353–368, 2011.

[6] M. Lakshmanan and D. Senthilkumar, *Dynamics of Nonlinear Time-Delay Systems*. Springer-Verlag, 2010.

[7] F. Atay, "Van der pol's oscillator under delayed feedback," *J. Sound and Vibration*, vol. 218, no. 2, pp. 333–339, 1998.

[8] K. Hu and K. Chung, "On the stability analysis of a pair of van der pol oscillators with delayed self-connection, position and velocity couplings," *AIP Advances*, vol. 3.

[9] https://www.zhdk.ch/index.php?id=icst_downloads, accessed: 2016-02-25.

Using Software Emulation to Explore the Creative and Technical Processes in Computer Music: John Chowning's *Stria*, a case study from the TaCEM project

Michael Clarke
CeReNeM
University of Huddersfield
j.m.clarke@hud.ac.uk

Frédéric Dufeu
CeReNeM
University of Huddersfield
f.dufeu@hud.ac.uk

Peter Manning
Music Department
Durham University
p.d.manning@durham.ac.uk

ABSTRACT

The TaCEM project (Technology and Creativity in Electroacoustic Music) has investigated the relationship between technological innovation and compositional processes on the basis of nine case studies, including John Chowning's Stria *(1977). Each case study involved researching the historical and contextual background of the work, emulating the technology used to create it and analyzing its musical structure. For each of these electroacoustic works, a specially designed software package has been developed, forming an important part of the project outcome. If* Stria*, as a classic work of the electroacoustic repertoire, has been much written about, the study presented in this article is distinctive in that the software enables to present the results of this research in an interactive and aural form: its users can engage directly with the structure of the work and the techniques and processes used by Chowning to compose it. This article presents this interactive aural analysis approach, its application to* Stria*, and the interactive resources embedded into the resulting software.*

1. INTRODUCTION

Researching music in which technology plays an integral part in the creative process presents particular challenges as well as opportunities for musicologists. This is particularly the case in works where technology has changed the way in which the music is conceived and where detailed knowledge of the technology therefore plays a crucial part in developing a full understanding of the creative process. The task becomes more difficult in cases the technology used to produce the original work no longer exists. But even where the technology does exist, it may not be easily available to a wide range of researchers or may not be in a form they will find accessible. Written documentation and description can give some indication of the technical system but is no substitute for the knowledge that can be gained from exploring a working version, trying out different options and hearing the results in a hands-on environment. And this is also especially important in connecting technical investigation with research into the musical structure of a work, exploring how the technical and creative interact. These relationships are crucial to a well-informed understanding of the works concerned, not least in terms of how the associated technologies have influenced the creative process. Although the importance of thus engaging with the *Technè* or art of bringing forth such works via a technical medium has been recognized for many years – see for example articles written by Di Scipio in 1995 [1] and Manning in 2006 [2] – the development of suitable tools for such investigations has still a long way to go.

In the TaCEM project, Technology and Creativity in Electroacoustic Music[1], a 30-month project funded by the Arts and Humanities Research Council in the UK, the authors have attempted to address these issues by creating interactive software to help investigate both the musical structure of works and the processes that led to their creation. This approach builds on Clarke's earlier work on Interactive Aural Analysis [3]. The main output of our project will be a book with substantial accompanying software. John Chowning's *Stria* is one of nine case studies examined for the project. Each case study involves researching the background to the work, emulating the technology used to create it and analyzing its musical structure[2]. With each of our case studies, a specially designed software package forms an important part of the outcome. *Stria* is a classic of the electroacoustic repertoire and, as such, much has previously been written about it, most notably in a 2007 edition of the *Computer Music Journal* [5, 6, 7, 8][3]. What makes our study distinctive is the way software enables us to conduct our research and present it from an interactive aural perspective, and in so doing explores the characteristics of the works being studied in greater depth. We can investigate technological aspects by working with models that emulate the processes employed and comparing our results with the original. We can examine the significance of the choices the composer made in shaping the work by trying out alternative parameter settings within their creative environment and evaluating the results aurally. We have drawn on previous work on *Stria* and have also benefitted greatly from the help and advice of John Chowning himself. Video interviews we conducted with the composer during a visit to Stanford form another important aspect of our software package, adding a poietic perspective. Another very significant aspect of our study, one not represented in this paper, is contextual research, placing the work in its wider technical and musical context. The purpose of this paper is not to give a full analysis of *Stria* (that will appear in the book arising from the project) but rather to provide an introduction to our working methods together with the software and to demonstrate and discuss the advantages of approaching this type of repertoire through the use of interactive resources.

[1] http://www.hud.ac.uk/research/researchcentres/tacem/ (last visited May 11, 2016).
[2] See for instance [4].
[3] See also [9].

2. OVERVIEW OF THE SOFTWARE

The software for our study of *Stria* incorporates various different components, combining technical study, musical analysis and video interviews with the composer. The associated text, in a book chapter, provides further contextual information and more detailed explanation of the technical and creative issues presented in the software. To gain most from these materials, the text and software need to be studied in tandem and the chapter will contain links to video demonstrations of the software to facilitate this articulation. The software package for *Stria* comprises seven interactive explorers, interleaved with related video extracts from our interviews with the composer. Figure 1 provides an overview of the TaCEM software for the study of *Stria*.

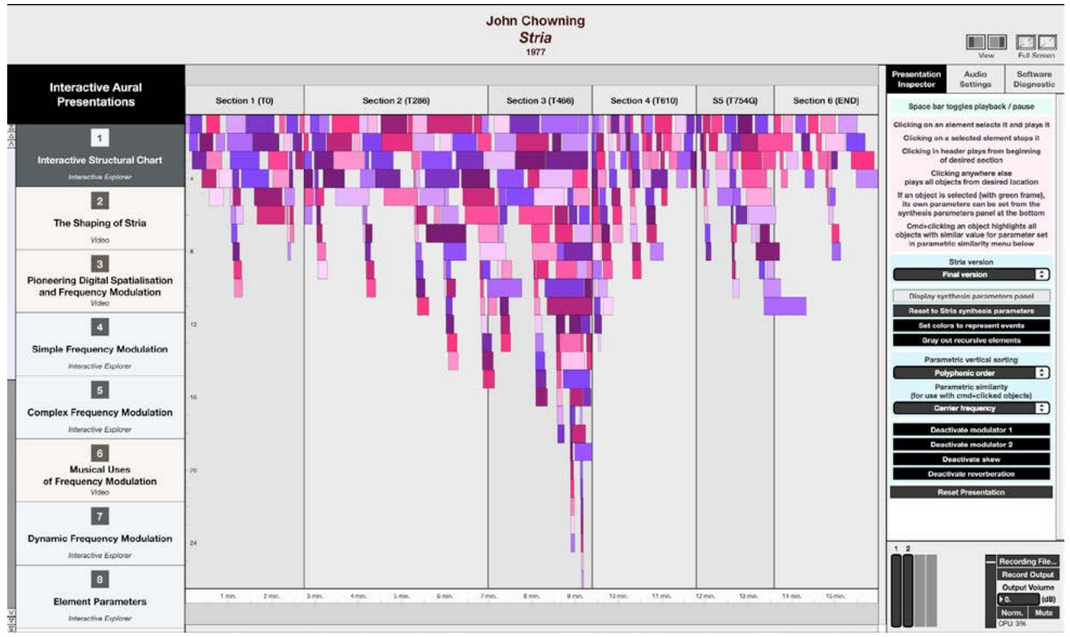


Figure 1. Overview of the TaCEM software for the analysis of *Stria*. On the left side is an Interactive Aural Presentations bar, enabling the user to navigate through interactive explorers and videos. The central canvas is the main interactive workspace for a given presentation (here, presentation 1: the Interactive Structural Chart representing the global structure of *Stria*). On the right side is the presentation inspector, providing access to further options for advanced visualization and playback.

3. INTERACTIVE EXPLORERS

The first interactive presentation is an Interactive Structural Chart. As seen in figure 1 above, this chart sets out all the "elements" that comprise *Stria* in temporal order from left to right along the horizontal axis. "Element" is the term Chowning uses for the smallest component of the work, roughly equivalent to a "note" in a traditional score. These elements come together to form the "events" which in turn build up the six "sections" of *Stria*. In our interactive structural chart, individual elements can be heard by clicking on them and, alternatively, the whole texture at any point in the work may be played. The criteria used to order the vertical arrangement of the elements can be changed by the user from a menu in the presentation inspector, on the right side of the window. The default option, as on figure 1, simply presents the elements in polyphonic order, so that overlapping elements are stacked vertically. However, any of the individual synthesis parameters used in the creation of *Stria* may determine the vertical arrangement. For example, the carrier frequency, either of the two modulation frequencies, the reverberation amount or the distance factor can be represented on the vertical axis. In this way, an overview of the shape of the work can be seen from many different perspectives. Importantly, because of the interactive aural nature of the chart, seeing can be linked directly to hearing. The sounds presented in this chart are synthesized live by the emulation software and this means that it is

possible in listening to examples from this chart to bypass certain aspects of the synthesis process so that their significance and contribution to the overall sound can be perceived and understood. The aspects that can be deactivated are the operation of the two modulators, the skew (a small dynamic pitch variation adding richness to the overall sound) and the reverberation. Interestingly, the shape of the work as shown (and heard) by this interactive chart, for example when ordered by carrier frequencies (figure 2), does not correspond neatly to the "V"-shape frequently portrayed, for example on the cover of the aforementioned 2007 issue of the *Computer Music Journal* (figure 3). It is not clear what specific data, whether parameter data inputted or outputted results, was used to draw this shape, though it does correspond to the composer's own description of the shape of the work (for example, in our interviews with him)[4]. In terms of the parameters shown in our Interactive Structural Chart, the form of the work, although generally following this shape, is a far more complex interaction with many contributing factors. Being able to identify and explore such issues is one of the advantages of working directly with an interactive emulation of the technology and with flexible visualization features.

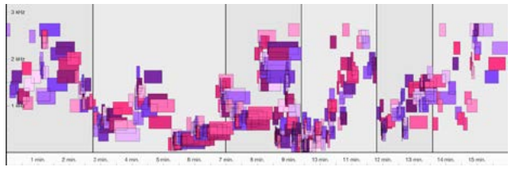


Figure 2. Elements of *Stria* sorted vertically by carrier frequencies

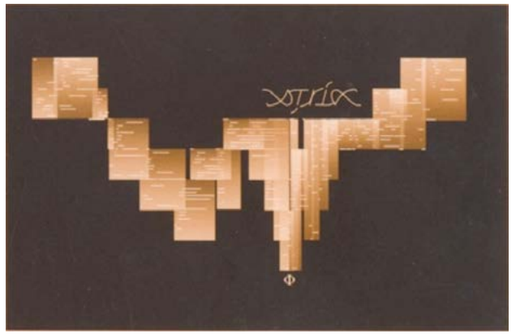


Figure 3. The shape of *Stria* as depicted on the cover of the 2007 issue of the *Computer Music Journal*

The next group of interactive explorers relate to the synthesis method used in the work. This is the frequency modulation (FM) method, famously developed by Chowning himself and later used in many different software packages and patented for use in commercial synthesizers by Yamaha. Three interactive explorers introduce the reader in stages to this synthesis method and how it is used in *Stria*. Firstly, the basic principles of FM are introduced followed by the more complex, two-modulator version of FM used in *Stria*. Our interactive software enables users to try out the technique for themselves either using examples of the input data from *Stria* itself or inputting their own settings. The software has an option to display numerical data about the frequencies generated by the modulation process (the frequencies of the sidebands), as well as showing them graphically (figure 4). Displaying this data numerically is useful in analyzing the often complex results arising from two-modulator FM synthesis. Indeed, it is particularly helpful in exploring the significance of the ratio used by Chowning in this work (the Golden Mean) and what this means in terms of the timbres generated.

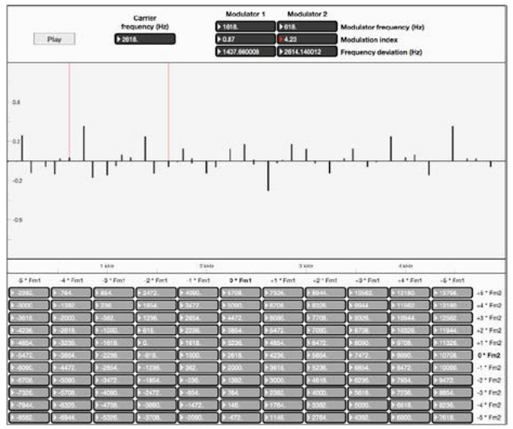


Figure 4. Two-modulator, one-carrier frequency modulation with graphical and numerical representations of sideband frequencies

The next interactive FM explorer demonstrates the potential of dynamic evolution of timbres in FM synthesis using envelopes to shape parameters (figure 5).

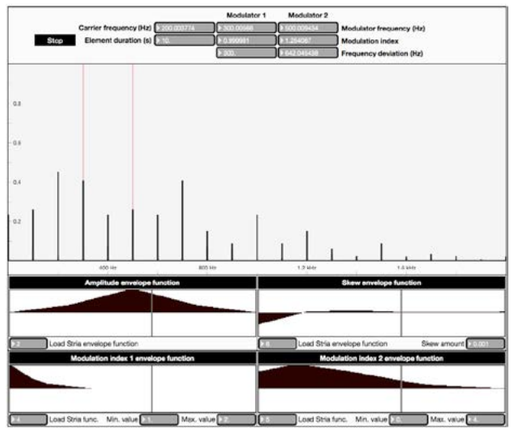


Figure 5. Interactive explorer of dynamic FM. The envelopes are applied to the overall amplitude, to the skew, and to the two modulation indexes.

[4] Another "V"-shape can be found in [10], p. 137. As Zattra notes in [6], p. 45, "Dodge and Jerse consider this [latter] figure a sketch of the shape (without any claim to be precise in details) of the 18-minute piece".

The software displays the envelopes and these can be changed interactively to facilitate greater understanding of their importance through direct aural experience.

A further interactive explorer builds on the FM examples already introduced and extends them to include all the parameters involved in defining an element in *Stria*, including spatialization. In this explorer, the user can shape individual elements using the same parameters as used to control the synthesis engine in *Stria*. These synthesis parameters are grouped according to the following categories: time parameters, frequency parameters, modulation parameters, spatialization, and envelope functions (figure 6). A two-dimensional surface enables to set the amplitudes of the four output channels by providing an angle directly. The envelope layout highlights Chowning's specific use of the terms "attack" and "decay" in *Stria*: the attack setting, in seconds, determines the time to go through the first quarter of the envelopes. Likewise, the decay setting determines the time to go through the last quarter of the envelopes. Interacting with this explorer reveals, by modifying the duration, attack and decay parameters and watching the playhead on the envelope panels, the non-linear indexing of Chowning's functions.

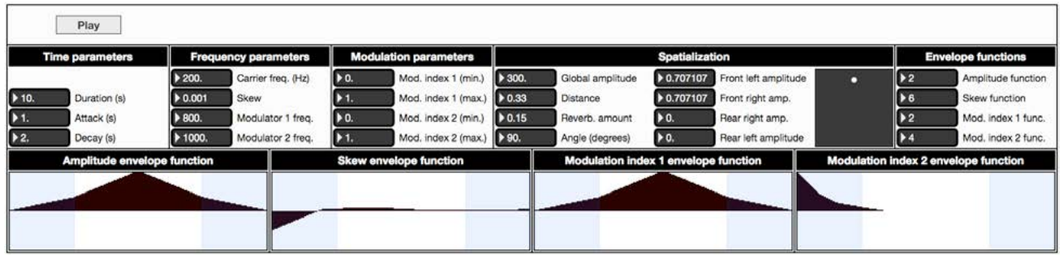


Figure 6. Interactive explorer for the definition of one element according to synthesis parameters

In the composition of *Stria* itself, Chowning did not shape the elements individually. Instead, the composer programmed an algorithm to do this work, with him inputting data to higher-level parameters he created to define the structure of a whole event. Unlike our emulation, *Stria* was produced using software that did not operate in real-time. Indeed, there were significant delays waiting for sounds to be generated. In our interviews, the composer described the advantages he found in working in this way: it gave him time to reflect and plan at a time when many of the ideas and techniques he was employing were new and unexplored. Furthermore, the piece was not produced directly in a single process but in two successive stages. The first stage, using the program the composer himself constructed in SAIL (Stanford Artificial Intelligence Language), used algorithms to generate the data defining the elements comprising each event in the work. To realize these elements in sound, a second stage was then required, in which the data generated by the SAIL algorithm was imported into Music 10 as score data and the sounds were then generated (again, not in real-time). The Music 10 orchestra for *Stria* remains fixed throughout the work and generates sounds using the two-modulator FM synthesis algorithm mentioned above, with the addition of global reverberation and four-channel spatialization.

In our software emulation of the whole process, these two separate stages are combined into one and the whole operation is run in real-time. Despite the benefits Chowning found in not working in real-time when composing the work, in the context of studying the music and its technical and creative processes, real-time has the advantage of allowing readers to engage more directly with the music as sound and helps develop an understanding of the aural significance of particular parameters and of the settings chosen by the composer. As a default, users are provided with the original settings Chowning used in *Stria*, but they may also input their own alternatives, either using an emulation of the composer's original interface, inputting data one item at a time in response to on-screen prompts (figure 7), or using a more modern graphical interface (figure 8).

Chowning's original SAIL program takes the user parameters and inputs the data from these into an algorithm that calculates the individual elements for a particular event. The algorithm is determinist: inputting the same data will always result in rigorously the same outcome.

To the novice user, the significance of the input parameters may not be immediately obvious, especially when encountered in the original command-line format. This is particularly the case as some aspects of the resulting sound (for example the fundamental pitch of an element) are influenced by the interaction of several different parameters (in the case of the fundamental frequency these include range, division of this range, multiplier, etc.). To make this easier to understand and to show the effect of changing parameter settings, our software emulation incorporates visual displays of the parameters and the resulting elements. Furthermore, to facilitate conceptual understanding of how Chowning's design of the algorithm enables him to shape the music, the emulation includes the option of showing visually how an event is formed step by step, starting with temporal definitions for events and the elements they contain (for the event: start time, duration, attack duration[5]; for the elements: start positions and durations), then frequency definitions (base frequency, frequency space), followed by modulation parameters, element attacks and decays, and spatialization. In this way, the creative and technical thinking behind the algorithm is revealed.

[5] In the terminology of *Stria* and its programs, the attack duration of an event is the duration within which new elements can be generated. If an event has a duration of 60 seconds and an attack duration of 30 seconds, no new element will appear during the second half of the event.

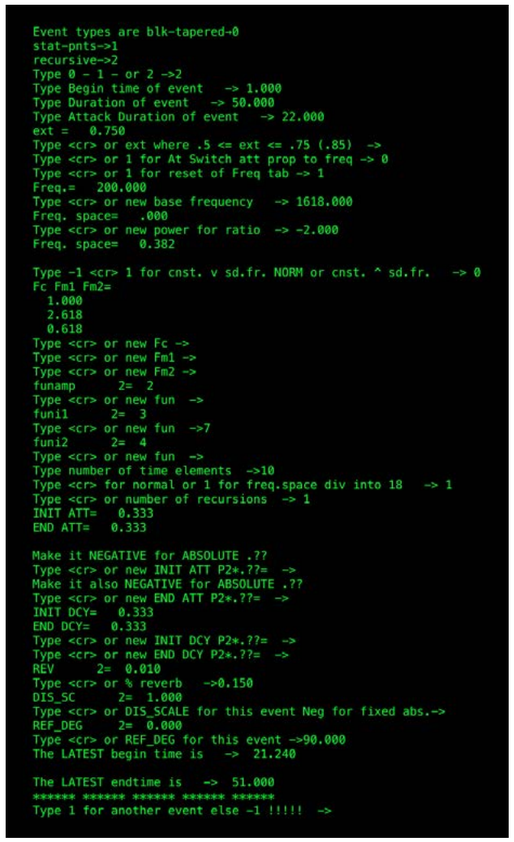


Figure 7. Emulation of the SAIL terminal interface to create one event in *Stria*

Figure 8. Graphical User Interface for the creation of one event

4. VIDEOS

Our analytical and technical investigations into *Stria* are complemented by poietic material in the form of interviews with the composer that we conducted over three days in March 2015. Extracts from these interviews are interleaved with the software presentations so that topics explored through the software appear alongside related discussions with the composer (figure 9). These interviews will also be referenced and in part transcribed in the accompanying book text.

The topics covered by the interviews range from specific issues relating to the composition of *Stria* to general discussion of the FM synthesis technique, to much broader topics concerning the Chowning's career and his more recent compositional concerns.

Figure 9. Filmed interview with John Chowning at CCRMA in Stanford (March 2015), as embedded amongst the presentations in the TaCEM software

In total there are nine videos of varying length. They are entitled according to the following topics: The shaping of *Stria*; Pioneering digital spatialization and Frequency Modulation; Musical uses of Frequency Modulation; Approaches to programming (SAIL and Max); The different versions of *Stria*; Encounters and interactions with Jean-Claude Risset; Academic and commercial impact of Chowning's work; Chowning's compositional process and career; and From *Stria* (1977) and *Phoné* (1981) to *Voices* (2005). In this way, the detailed and specific study of *Stria* and the technology behind it can be related to the composer's creative intentions and to the broader picture of his contribution to the development of computer music.

5. CONCLUSION

Combining software with written text and filmed interviews enables those who use our resources to gain a deeper understanding of a work, in this case *Stria*, than would be possible using text alone:

- they can discover the potential of synthesis techniques by working with them rather than just reading about them as theory;
- they can explore the musical shape of works as sound;
- they can hear the musical impact of the choices the composer made;
- they can see and hear composers giving their own accounts of the works and their broader context.

Especially as, over time, many of the original technical resources employed in particular works become obsolete, our approach of creating good approximations to these technologies helps to ensure that detailed understanding of those technologies and of what it was like to work with them is preserved. Furthermore, the articulation of software, text and interviews helps to preserve and transmit to new generations lessons learnt about the successful combining of technical knowledge and creative inspiration. Indeed, in trialing these materials in a pedagogic context we have discovered their potential for bringing together aspects of music technology teaching that are more often kept as isolated units: students can learn about the history of computer music, they can explore techniques (e.g. FM synthesis), they can investigate how a particular work is structured musically and, using our emulation software if they wish, they can try out creative ideas inspired by all that they have learnt, producing their own compositional sketches. In a digital age, surely it does not make sense to rely solely on written text to try and convey matters relating to sound, complex technology, and creativity. Engaging with this repertoire aurally and interactively adds an important additional dimension to the mode of enquiry and significantly enriches what can be conveyed and learnt.

Acknowledgments

The research presented in this project is part of the TaCEM project, funded by the United Kingdom's Arts and Humanities Research Council (AHRC). The authors would like to thank John Chowning for his generous help in investigating *Stria* and his broader creative process.

6. REFERENCES

[1] A. Di Scipio, "Centrality of *Techné* for an Aesthetic Approach on Electroacoustic Music", *Journal of New Music Research*, vol. 24, no. 4, 1996, pp. 369-383.

[2] P. Manning, "The significance of *Techné* in Understanding the Art and Practice of Electroacoustic Composition", *Organised Sound*, vol. 11, no. 1, 2006, pp. 81-90.

[3] M. Clarke, "Analysing Electroacoustic Music: an Interactive Aural Approach", *Music Analysis*, vol. 31, no. 3, 2012, pp. 347-380.

[4] M. Clarke, F. Dufeu, P. Manning, "From Technological Investigation and Software Emulation to Music Analysis: An integrated approach to Barry Truax's *Riverrun*", *Proceedings of the 2014 International Computer Music Conference / Sound and Music Computing conference*, Athens, 2014, vol. 1, pp. 201-208.

[5] M. Meneghini, "An Analysis of the Compositional Techniques in John Chowning's Stria", *Computer Music Journal*, vol. 31, no. 3, 2007, pp. 26-37.

[6] L. Zattra, "The Assembling of *Stria* by John Chowning: A philological investigation", *Computer Music Journal*, vol. 31, no. 3, 2007, pp. 38-64.

[7] K. Dahan, "Surface Tensions: Dynamics of *Stria*", *Computer Music Journal*, vol. 31, no. 3, 2007, pp. 65-74.

[8] O. Baudouin, "A Reconstruction of *Stria*", *Computer Music Journal*, vol. 31, no. 3, 2007, pp. 75-81.

[9] B. Bossis, "*Stria* de John Chowning ou l'oxymoron musical : du nombre d'or comme poétique", in É. Gayou (ed.), *John Chowning. Portraits Polychromes*, Paris, Ina-GRM, TUM-Michel de Maule, 2005, pp. 87-105.

[10] C. Dodge, T. Jerse, *Computer Music. Synthesis, Composition, and Performance*, 2nd ed., London and New York, Schirmer, 1997.

CONCATENATIVE SYNTHESIS VIA CHORD-BASED SEGMENTATION FOR *AN EXPERIMENT WITH TIME*

Daniele Ghisi
STMS Lab
(IRCAM, CNRS, UPMC)
Paris, France
danieleghisi@bachproject.net

Mattia Bergomi
Champalimaud Neuroscience Programme
Champalimaud Centre for the Unknown
Lisbon, Portugal
mattia.bergomi@neuro.fchampalimaud.org

ABSTRACT

In this article we describe a symbol-oriented approach to corpus-based electroacoustic composition, used during the writing of the sound and video installation An Experiment with Time*, by one of the authors. A large set of audio files (picked from a database spanning the whole history of western music) is segmented and labeled by chord. Taking advantage of the* bach *library for Max, a meta-score is then constructed where each note actually represents an abstract chord. This chord is potentially associated to the whole collections of grains labeled with it. Filters can be applied in order to limit the scope to some given subset of sound files or in order to match some specific descriptor range. When the score is rendered, an appropriate sequence of grains (matching the appropriate chords and filters) is retrieved, possibly ordered by some descriptor, and finally concatenated via standard montage techniques.*

1. INTRODUCTION

Corpus-based concatenative synthesis [1] is a largely known technique, providing mechanisms for real-time sequencing of grains, according to their proximity in some descriptors space. Grains are usually extracted from a corpus of segmented and descriptors-analyzed sounds. Composing via a corpus, in a way, allows composers to take a 'step back' from the work itself — in some sense, ordering, clustering, filtering become the very first compositional acts. When brought to the extreme consequences, this attitude yields some of the most intriguing piece of arts, such as Jonathan Harvey's automatic orchestrations in *Speakings* or Christian Marclay's montage in *The Clock*.

Among the existing tools dealing with audio corpus-based concatenative synthesis, *CataRT* [2] is probably the most widely used. Taking advantage of the features in the FTM [3] and (more recently) *MuBu* [4] libraries, it provides tools for sound segmentation and analysis, as well as for the exploration of the generated corpus via an interactive two-dimensional display, both inside the Max environment and as a standalone application.

However, neither *CataRT* standalone application nor *CataRT*'s *MuBu* implementation easily allow a chord-based segmentation, which should be carried out in some other way, and then imported. More importantly, neither of these pieces of software offers tools to handle symbolically notated music (i.e. 'notes', rather than 'sounds'), so that the traditional composer's experience with such tools is often limited to the domains of discovery and improvisation. In other words, *CataRT* is mostly oriented to real-time performance and live interaction, omitting essentially any symbolic representation of events (except for the very crude *piano roll* display provided with *MuBu*). And yet, today, few composers are willing to give up symbolic writing [5].

This counterposition recalls the one between 'performative' and 'compositional' paradigms, that Puckette introduced in [6]. Recently, some work has been done in order to equip corpus-based concatenative synthesis techniques with symbolic notation (see, for instance, [7]), and more generally, it seems that the whole computer-assisted composition community is making a conjoint effort to narrow the gap between 'performative' and 'compositional' paradigms (see for instance the *bach* project [8, 9], and OpenMusic's 'reactive' mode [10]).

Continuing on this path, in this article we describe a symbolic approach to corpus-based electroacoustic composition, used by one of the authors during the writing of the sound and video installation *An Experiment with Time* [1]. This approach mainly relies on a large set of audio files (picked from a wide database spanning the whole history of western music), segmented and tagged by chord. This allows the composer to operate on a meta-score, where each note actually represents an abstract chord; the score in turns can be rendered via concatenative synthesis of chord-labeled samples (either in real-time or off-line).

2. DATABASE AND SEGMENTATION

The database for *An Experiment with Time* is composed by about 3000 tracks of classic, contemporary, rock, pop and jazz music, sampled from the whole history of western music. The harmonic transcription of each song has been computed using the algorithm presented in [11]. This effective algorithm allows to set a specific dictionary, in order to select a certain number of chord classes. Each element of the dictionary is indicated by specifying both the bass and chord relative pitch classes. Thus it is possible, for instance, to associate to a major chord only its root form or identify it with its inversions. In the case of *An Experiment with Time* the chord dictionary was defined in order to detect the classes listed in Table 1.

[1] www.anexperimentwithtime.com

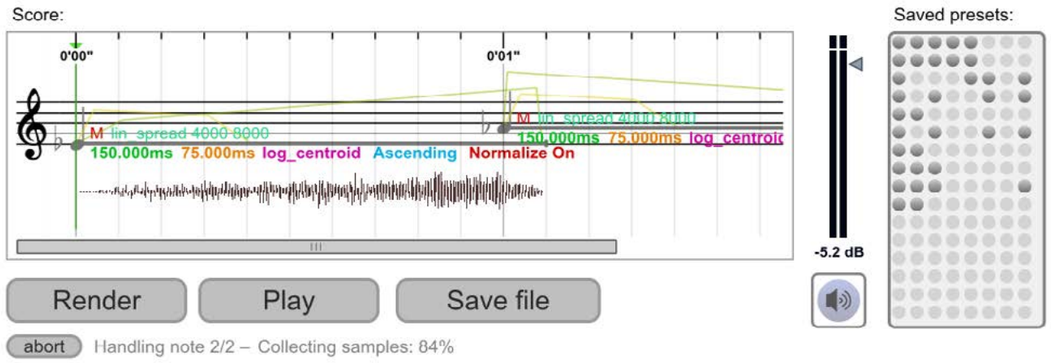


Figure 1. Interface of the off-line composition module.

chord class	*pitch class structure*
N.C.	no chord
maj	(0, 4, 7)
maj/3	(0, 3, 8)
maj/5	(0, 5, 9)
aug	(0, 4, 8)
min	(0, 3, 7)
dim	(0, 3, 6)
6	(0, 4, 7, 9)
7	(0, 4, 7, 10)

Table 1. Chord classes in *An Experiment with Time*.

This particular choice of chord classes has been done in order to include the four standard tonal trichords (major, minor, augmented and diminished) and a few of their basic variants.

Thereafter each audio track has been associated to a JSON file specifying both the onset of the chord in seconds and its class as follows

```
{"chords":[
{"position":0,
"chordname":"N.C.",
"avbpm":139},
{"position":230574.149659,
"chordname":"B\/F#",
"avbpm":139},...
]}
```

Finally, each audio file has been cropped in *harmonic grains* according to these features. This procedure allowed us to create a new database organized in folders named with a specific pair $(root, class)$ and containing harmonic grains labelled as $chordname\_n\_path\_title$. The file path has been preserved in order to facilitate queries involving the names of the folders containing specific files. The natural number n represents the position of the chord with respect to the entire harmonic sequence of the audio track.

3. COMPOSITION MODULES

3.1 Off-line composition module

The most natural way to exploit the database segmented by chord, is to compose with chords instead of notes. If we limit ourselves to major and minor chords, this perspective is perfectly dual; for instance one can easily switch back and forth from the classic Tonnetz to its dual, containing major and minor chords (as shown, e.g., in [12]).

A very handy tool for composing with notes carrying some additional meaning is the *bach* library for Max. The *bach* library brings symbolic notation in a real-time environment [9]; in particular, each note carries additional meta-information structured in 'slots'; such information can be shaped in various forms [7].

For our purposes, we have set up a proportionally notated score (see Fig. 1), so that each note stands for the fundamental of a chord, whose type is specified via the first slot. This representation is handier than having to specify all the voices of the chord, as it allows to separately control the two orthogonal and pertinent parameters: fundamental and chord type.

For each note, additional slots carry the information for the concatenative rendering: the duration of each unit, the inter-unit distance, as well as the descriptor according to which the units should be sorted (if any) and the sorting direction. Another slot narrows the choice of units so that a certain descriptor value lies in a given range; furthermore, additional slots provide amplitude envelopes (both for each unit and for the whole sequence of units). Finally, a slot is dedicated to filtering the database according to words or parts of words appearing in the file name or path; this is an extremely quick and effective way (via Unix find command) to narrow the search to a tag or a combination of tags (e.g. 'Mozart', or 'Symphony', ...). All slots and their usages are shown in Table 2.

The score is by default rendered in off-line mode: note are processed one by one. For each note, three stages of rendering are necessary: the random retrieval of the sound files satisfying the input criteria (chord, descriptor range, tags); the sorting of such sound files depending on the value of the specific descriptor (if any); the montage of random portions (units) of the sound files into a single

slot number	*slot content*
1	chord class
2	grain duration
3	grain distance
4	grain amplitude envelope
5	global note amplitude envelope
6	filter by descriptor ranges
7	sort by descriptor
8	sorting direction
9	grain normalization

Table 2. Slots setup for the offline composition module.

sequence, associated with the note. The process can be aborted at any time. Once the score is rendered, the corresponding waveform appears below the score, and stays graphically synchronized with it. The score can be then played, saved, or rendered anew.

3.2 Real-time module

The concatenative process described in section 3.1 can also be accomplished in a real-time way, although the retrieval of sound files can take a non-negligible time when switching chords, depending on the number of segmented files per chord. One might use the real-time capabilities as an exploratory tool, before turning to the off-line module for actual symbolic writing.

3.3 Chord-sequence module

The off-line composition module, as described in section 3.1, randomly concatenates units of sound files for each note. The inner coherence for the sequence somehow relies on the fact that all units share the same harmonic content, and on the fact that units are sorted according to a specific descriptor. However, no particular continuity is guaranteed when notes change.

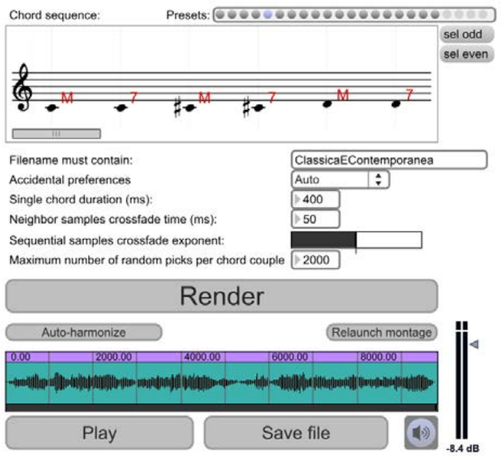


Figure 2. Interface of the chord-sequence module.

A specific module has been developed to allow chord sequences to be rendered more smoothly (see Fig. 2): the user defines a chord sequence in a similar manner as of section 3.1 (notes represent chord fundamentals and carry meta-information). In this case, however, couples of contiguous chords are rendered at once. For each couple of chords, the algorithm searches for a sound file where such chords show up exactly in the correct order, without discontinuity; this is made possible by the fact that the segmentation process retains in the output name a chord index. All the overlapping couples of chords are then cross-faded in order to create the complete sequence. In this case, a note does not represent a sequence of units, rather a single unit, which on the other hand is guaranteed to join seamlessly with the previous and following one.

A set of basic synthesis parameters can be customized by the user, and an auto-harmonize button is set in place to automatically detect chord types depending on the notes of the sequence (this is especially useful when harmonizing scales).

4. *AN EXPERIMENT WITH TIME*

The paradigms and tools described in the previous sections are tailored for the audio and video installation *An Experiment with Time*, by one of the authors [2]. This work inspired by the eponym book by John W. Dunne, depicts the writing of a diary, where the main character carries out an experiment on his own dreams, and proposes different hypothesis on the nature of time.

The video loop, during 46 minutes, represents the writing of a diary during a whole year (from January to December). The starting point for the musical writing is a straightforward association between months and major chords, so that the whole year loop is handled like a sequence of perfect cadences in the tempered cycle of fifths (January being B major, February being E major, and so on, till December being F# major, and hence looping). Although the internal handling of the musical material becomes more complex (different chord types are explored and different fundamentals are used occasionally to underline specific passages), everything in the piece is conceived with respect to this simple sequence, which thus represents the skeleton of the whole musical loop.

The starting point for *An Experiment with Time* is the corpus of segmented audio files described in section 2. This corpus has been chosen so that time can be set a parameter of the corpus itself. The relation between the historical time and the musical time is powerful enough to create interesting diffraction patterns. As an example, during June, a radio broadcasts some sort of 'history of C major' [3], composed by C major samples ordered with respect to their composition year. Similar processes are used diffusely throughout the whole work.

The chord-sequence module, as described in section 3.3, is often used to produce chord sequences undergoing extremely simple rules, such as a tremolo-like alternation between major and minor chords (Fig. 4), or a sequence

[2] The installation was premiered in Paris, festival *Manifeste*, the 1st june 2015. A live version of the work, for ensemble, video and electronics, has been premiered in January 2016. A teaser, as well as some excerpts, are available on the official website www.anexperimentwithtime.com.

[3] An officer named Major C. is also a supporting character in the video, hence the word play.

Figure 3. A frame from *An Experiment with Time*.

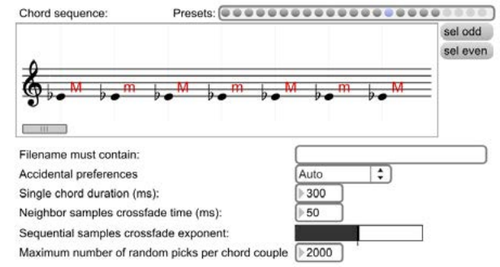


Figure 4. A tremolo-like alternation between major and minor chords. During rendering, only transitions between E flat major and minor chords (and vice versa) are retained from the database; for each couple of notes, a transition is chosen and then the sequence of transitions is created via crossfades.

of continuous, smooth deceptive cadences (Fig. 2). This latter, at the end of December, flows into a continuous Risset-like glissando composed by a micro-montage of small chunks of vocal glissandi [4].

5. CONCLUSIONS AND FUTURE WORK

This line of research and the framework that has been described have proven to be very fruitful, since they provide the composer both with a strong control on harmonic content and with the possibility to operate symbolically upon it.

Two improvements should be considered. On one side, chord-based concatenative synthesis (as described in the module of section 3.1) should be provided with beat-alignement capabilities, in order to preserve some sort of rhythmic grid or pattern throughout the sequence of units. On the other hand, the continuity of couples of neighbor chords in the module of section 3.3 might be extended to overlapping subsequences of an arbitrary number N of chords, to guarantee an even smoother continuity.

6. REFERENCES

[1] D. Schwarz, "Corpus-based concatenative synthesis," *Signal Processing Magazine, IEEE*, vol. 24, no. 2, pp. 92–104, 2007.

[2] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-Time Corpus-Based Concatenative Synthesis with CataRT," in *Proceedings of the International Conference on Digital Audio Effects*, 2006, pp. 279–282.

[3] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Müller, "FTM - Complex Data Structures for Max," in *Proceedings of the International Computer Music Conference*, 2005.

[4] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi, "MuBu & Friends - Assembling Tools for Content Based Real-Time Interactive Audio Processing in Max/MSP," in *Proceedings of the International Computer Music Conference*, Montreal, Canada, 2009.

[5] F. Lévy, *Le compositeur, son oreille et ses machines à écrire*. Vrin, 2014.

[6] M. Puckette, "A divide between 'compositional' and 'performative' aspects of Pd," in *Proceedings of the First Internation Pd Convention*, Graz, Austria, 2004.

[7] A. Einbond, C. Trapani, A. Agostini, D. Ghisi, and D. Schwarz, "Fine-tuned Control of Concatenative Synthesis with CataRT Using the bach Library for Max," in *Proceedings of the International Computer Music Conference*, Athens, Greece, 2014.

[8] A. Agostini and D. Ghisi, "Real-time computer-aided composition with *bach*," *Contemporary Music Review*, no. 32 (1), pp. 41–48, 2013.

[9] ——, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal*, vol. 39, no. 2, pp. 11–27, 2015/10/03 2015. [Online]. Available: http://dx.doi.org/10.1162/COMJ_a_00296

[10] J. Bresson, "Reactive Visual Programs for Computer-Aided Music Composition," in *IEEE Symposium on Visual Languages and Human-Centric Computing*, Melbourne, Australia, 2014.

[11] M. Mauch, "Automatic chord transcription from audio using computational models of musical context," Ph.D. dissertation, School of Electronic Engineering and Computer Science Queen Mary, University of London, 2010.

[12] D. Tymoczko, "The Generalized Tonnetz," *Journal of Music Theory*, vol. 56, no. 1, pp. 1–52, 2012.

[4] The glissando detection and segmentation was carried out by hand.

Spatiotemporal Granulation

Muhammad Hafiz Wan Rosli
Media Arts & Technology Program,
University of California, Santa Barbara
hafiz@mat.ucsb.edu

Curtis Roads
Media Arts & Technology Program,
University of California, Santa Barbara
clang@mat.ucsb.edu

ABSTRACT

This document introduces a novel theory & technique called Spatiotemporal Granulation. Through the use of spatially encoded signals, the algorithm segments temporal and spatial information, producing grains that are localized in both time, and space. Well- known transformations that are derived from classical granulation, as well as new manipulations that arise from this technique are discussed, and outlined. In order to reassemble the grains into a new configuration, we explore how granulation parameters acquire a different context, and present new methods for control. We present findings, and limitations of this new technique, and outline the potential creative and analytical uses. The viability of this technique is demonstrated through a software implementation, named "Angkasa".

1. INTRODUCTION

The process of segmenting a sound signal into small grains (less than 100 ms), and reassembling them into a new time order is known as granulation [1]. Many existing techniques can articulate the grains' spatial characteristics, allowing one to choreograph the position and movement of individual grains as well as groups (clouds). This spatial information, however, is generally synthesized, i.e. artificially generated. This stands in contrast to temporal information that can be extracted from the sound sample itself, and then used to drive resynthesis parameters.

Ambisonics is a technology that represents full-sphere spatial sound (periphonic) information through the use of Spherical Harmonics. This research aims to use spatial information extracted from ambisonics recordings as a means to granulate space.

By extracting this spatial information, the proposed method creates novel possibilities for manipulating sound. It allows the decoupling of temporal and spatial information of a grain, making it possible to independently assign a specific time and position for analysis and synthesis.

1.1 Motivation

Classical granulation (temporal segmentation) segments a one dimensional signal into grains lasting less than 100 ms, and triggers them algorithmically. These grains are then spatialized using a number of known techniques, described in Section 1.2. As opposed to the means of artificially generating the grains' spatial information, we are interested in exploring the possibilities of extracting grains from different positions in space.

By granulating (segmenting) the spatial domain, in addition to the temporal domain of a captured signal, we can extract grains that are localized in space and time. The ability to do so allow us to reassemble the grains in a new spatial and temporal configuration, as well as introduce a range of possibilities for transformation.

1.2 Related Work

The analysis, and extraction of grains from different positions in space is a research area that has yet to be explored. However, there has been a number of techniques used to position sound particles in space (spatialization).

Roads outlines the techniques used for spatialization of microsound into two main approaches [1]:

1. Scattering of sound particles in different spatial locations and depths
2. Using sound particles as spatializers for other sounds via granulation, convolution, and intermodulation

Truax, on the other hand, uses granular synthesis as a means to diffuse decorrelated sound sources over multiple loudspeakers, giving a sense of aural volume [2]. Barrett has explored the process of encoding individual grains' spatial information via higher-order Ambisonics, creating a virtual space of precisely positioned grains [3] .

The techniques outlined above aims to position grains in a particular location in space– spatialization. On the other hand, Deleflie & Schiemer proposed a technique to encode grains with spatial information extracted from an Ambisonics file [4]. However, this technique implements temporal segmentation, i.e. classical granulation, and imbues each grain with the component signals of the captured sound field.

In contrast, our method of spatiotemporal granulation segments the space itself, in addition to time, to produce an array of grains, localized in azimuth (θ), elevation (ϕ), and time (t).

2. THEORY

The classical method of granulation captures two perceptual dimensions: Temporal-domain information (starting time, duration, envelope shape), and Frequency-domain information (the pitch of the waveform within the grain and the spectrum of the grain) [1].

The described method granulates space, and adds another dimension to this representation: Spatial-domain information. The fundamental premise of this method lies in the extraction of spatial sound information, and the segmentation of this space into grains which are localized in spatial position, and temporal position. These grains will henceforth be individually referred to as a *Spatiotemporal grain* (Figure 1).

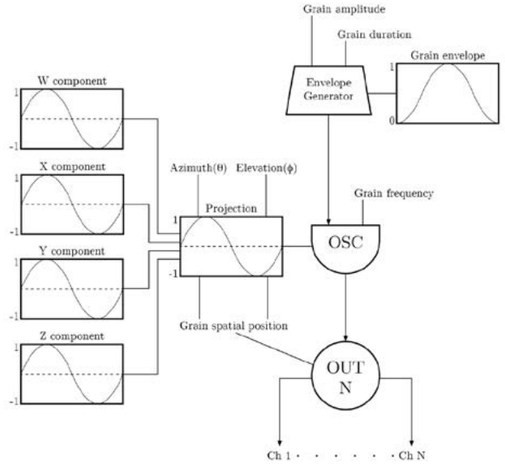


Figure 1: Block diagram of a basic spatiotemporal grain generator

Once the spatial information is decomposed (Section 2.2) into individual spatiotemporal grains, various manipulations can then be applied to transform the original sound field, described in Section 4.

2.1 Encoding of Spatial Sound

There are several microphone technologies that allow the capturing of spatial information, such as *X-Y/ Blumlein Pair*, *Decca Tree*, and *Optimum Cardioid Triangle*. However, these technologies do not capture the complete full-sphere information of spatial sound.

On the other hand, *Ambisonics* is a technique that captures periphonic spatial information via microphone arrays, such as the "SoundField Microphone" [5]. It is important to note that using this technique, sounds from any direction are treated equally, as opposed to other techniques that assumes the frontal information to be the main source, and other directional information as ambient sources.

The spatial soundfield representation of *Ambisonics* is captured via "Spherical Harmonics" [5]. Spatial resolution is primarily dependent on the order of the Ambisonics signal, i.e. order of Spherical Harmonics (Figure 2).

A first-order encoded signal is composed of the sound pressure W(Eq. 1), and the three components of the pressure gradient X(Eq. 2), Y(Eq. 3), Z(Eq. 4), representing the acoustic particle velocity. Together, these approximate the sound field on a sphere around the microphone array.

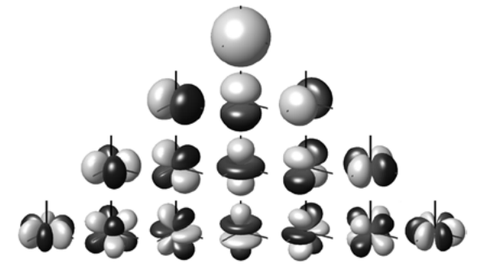


Figure 2: Visual representation of Spherical Harmonics up to third order [6].

$$W = \frac{1}{k}\sum_{i=1}^{k} S_i(\frac{1}{\sqrt{2}}) \quad (1)$$

$$X = \frac{1}{k}\sum_{i=1}^{k} S_i(\cos\phi_i \cos\theta_i) \quad (2)$$

$$Y = \frac{1}{k}\sum_{i=1}^{k} S_i(\sin\phi_i \cos\theta_i) \quad (3)$$

$$Z = \frac{1}{k}\sum_{i=1}^{k} S_i(\sin\theta_i) \quad (4)$$

2.2 Decoding of Spatial Sound

One of the strengths of Ambisonics is the decoupling of encoding (microphone & virtual), and transmission processes. This allows the captured sound field to be represented using any type of speaker configuration.

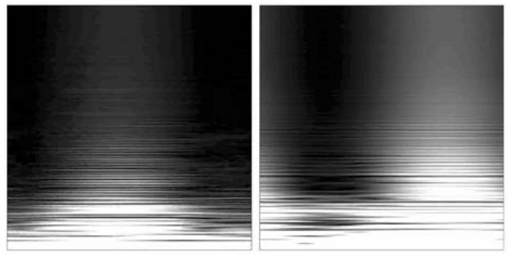
In practice, a decoder projects the Spherical Harmonics onto a specific vector, denoted by the position of each loudspeaker θ_j. The reproduction of a sound field without height (surround sound), can be achieved via Eq. 5.

$$P_j = W(\frac{1}{\sqrt{2}}) + X(\cos(\theta_j)) + Y(\sin(\theta_j)) \quad (5)$$

2.3 Spherical Harmonics Projection

Consider the case where we have N number of loudspeakers arranged in a circle (without height). In the case where N is 360, each speaker is essentially playing back sounds to reconstruct the captured sound field at 1 degree difference. Instead of playing the sounds from 360 loudspeakers, we can use the information as a means to specify different sounds from different locations.

This forms the basis for extracting sound sources in space for the spatiotemporal grains. However, segmentation of the spatial domain can be increased to any arbitrary value, limited by the spatial resolution, outlined in Section 3.3. If we were to look at the frequency content of these extracted grains in the same temporal window (Figure 3), we can deduce that each spatially localized grain contains a unique spectrum. Additionally, the directionality of the particular sound object can also be estimated.



(a) Start time (sample): 39424 (b) Start time (sample): 50688

Figure 3: X-Axis= Azimuth (0°- 360°), Y-Axis= Frequency bin, Intensity= Magnitude of bin, Window size= 512

2.3.1 *Periphonic Projection*

Equation 5 can also be extended to include height information, i.e. extracting every spatiotemporal grain in space (Eq. 6).

$$P_j = W(\frac{1}{\sqrt{2}}) + X(\cos(\theta_j)\cos(\phi_j)) + Y(\sin(\theta_j)\cos(\phi_j)) + Z(\sin(\theta_j)) \quad (6)$$

The result of this decomposed sound field can be represented as a 2 dimensional array (azimuth & elevation) of spatiotemporal grains, in the same temporal window (Figure 4).

(a) Start time (sample): 39424 (b) Start time (sample): 50688

Figure 4: X-Axis= Azimuth (0°- 360°), Y-Axis= Elevation (0°- 360°), Intensity= Energy of localized spatiotemporal grain, Window size= 512

Each snapshot of time represents one spatiotemporal frame (Figure 4). By successively lining up these temporal "slices" (frames), we gain a representation of the full decomposition, i.e. every spatiotemporal grain in space & time.

3. ANALYSIS

3.1 Data Collection

The data used in this research are captured using the *SoundField ST350 surround microphone*, and processed on a *2015 Mac Pro (OSX 10.10.5)* via an *RME Fireface UFX* audio interface. The dataset was gathered from various venues around Europe (Figure 5), and the United States.

We would like to extend our gratitude towards *Staatliche Hochschule für Gestaltung* (Karlsruhe University of Arts and Design), and *Zentrum für Kunst und Medientechnologie* (Center for Art and Media), Karlsruhe, Germany for the use of equipment.

(a) Captured space (b) Close up of the ST350

Figure 5: ZKM Lichtof

Additionally, B-Format files were also downloaded from www.ambisonia.com, and www.ambisonic.info. Plans to capture using Higher-order microphone arrays are already in progress.

3.2 Initial Explorations

Initial explorations to determine the viability of this technique were realized using python, in particular the interactive ipython notebook. Decomposition of Ambisonic files (Section 2.3.1) were explored to visually parse (Figures 3, 4) the spectrum of spatiotemporal grains.

As the analysis proved that the segmentation of space creates spatiotemporal grains with unique spectral contents, it soon became necessary to acoustically verify the theory. An exploration in Max/ MSP soon followed, and evidently, the grains indeed *sound* different from one another, at 1 degrees difference in azimuth & elevation.

However, Max/ MSP soon proved to be a limited solution, due to the inherent limitation that control data is not processed as often as signal data. There are various ways to overcome this issue, such as reducing the defined block size, and using sample-accurate trigger externals. However, we have chosen to move away from this environment, not only due to the described limitation, but also to have a more suitable platform to handle real-time visualization (Section 6).

3.3 Spatial Resolution

Through analysis via spatiotemporal granulation, we have deduced that the spatial resolution is dependent on the following:

1. The *order* of the microphone array (Spherical Harmonics)
2. The *characteristics* of the captured signal (short transient versus long sustained sounds)
3. The *space* where the signal is captured in (wet reverberant hall versus dry open space, or acoustically treated spaces)

4. TRANSFORMATION

In this section, we outline, and discuss the possible transformations that can be applied to the spatiotemporal grains. The transformations are not limited to the ones described here. Rather, these are merely starting points, and potentially every single transformation that can be applied to classical granulation, could also be applied to spatiotemporal granulation. In addition, the extraction of spatial information adds unique new effects to these well-known transformations. Examples of such transformations include:

4.1 Per grain Transformations

Transformations performed on a "per grain basis", such as per grain reverberation, and per grain filtering, can be applied spatially. For instance, we can apply a type of convolution reverb to grains extracted from a certain direction ($\theta = 0°$ to $360°, \phi = 90°$).

Another example is to apply a bandpass filter to a grain with the highest energy at that time frame. In addition, one could shift the center frequency of each neighboring spatiotemporal grains' bandpass filter to create a secondary "spatial filtering" effect.

4.2 Granular Substitution

As stated in Section 3.3, the spatial resolution of this technique is greatly dependent on a few factors, including the order of the microphone array used to capture the sound field (in the case of recorded samples). This translates to how similar a spatiotemporal grain's spectrum is to its adjacent neighbor. On a first order microphone array, the resulting spatiotemporal grains may be highly correlated.

Granular Substitution allows us to "compose" the spatiotemporal frame in order to create a more interesting palette. The grains for Granular Substitution can be selected via different techniques, similar to those described in Section 5.1.1. It allows us to substitute selected spatiotemporal grains with other grains from a different spatial, or temporal position. Additionally, the grains can be substituted with other grains from a completely different spatial (or non spatial) sound recording.

4.3 Dictionary Based Methods

As an extension to granular decomposition via Dictionary Based Methods [7], spatiotemporal granulation can be incorporated to generate sparse approximations of atoms that are localized in time, frequency and space. Transformations such as morphological filtering (i.e. filtering tailored to specific sound structures), jitter, and mutation (granular morphing/ sonic metamorphosis) could give rise to a new family of transformations that affects temporal, frequency, and spatial domains.

4.4 Affine Transformations

Affine transformations such as translate, scale, rotate, shear, and reflect can be performed on the spatiotemporal frames (Figure 6), or the Spatial Read Pointer (Figure 7), as discussed in Section 5.1.2.

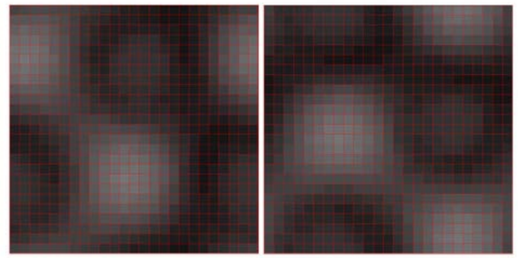


Figure 6: Left: Original spatiotemporal frame, Right: Transformed spatiotemporal frame

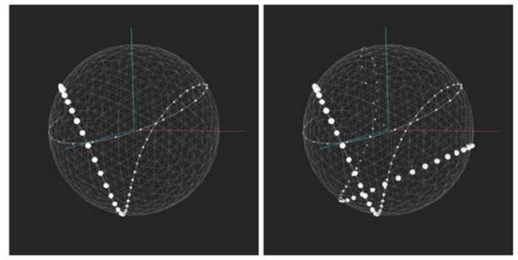


Figure 7: Left: Original Spatial Read Pointer, Right: Transformed Spatial Read Pointer

5. SYNTHESIS

Potentially every parameter used for classical granulation can be applied to spatiotemporal granulation. Roads [8] outlines the parameters that affect granulation as:

1. Selection order- from the input stream
2. Pitch transposition of the grains
3. Amplitude of the grains
4. Spatial position of the grains
5. Spatial trajectory of the grains
6. Grain duration
7. Grain density- number of grains per second
8. Grain envelope shape
9. Temporal pattern- synchronous or asynchronous
10. Signal processing effects applied on a grain-by-grain basis

Here, we discuss a few of these parameters that acquire a different context through spatiotemporal granulation:

5.1 Selection Order

The selection order is expanded to include not only the 1 dimensional selection order in time, but also the spatially encoded layout of the captured sound field. Previous methods of selection are also applicable in the spatial dimension:

5.1.1 *Selective Granulation*

Specifying only a certain region to be granulated allows us to selectively granulate the sound field. For example, we are able to granulate only 1 quadrant, and temporally stretch the grains that fall within that area, while allowing the other quadrants to progress at a different time speed.

Selection of these areas can be realized via different techniques, including (but not limited to):

1. Regions of the frame, such as quadrants, or sections
2. User defined selection of grains [9]
3. Statistical algorithms for quasi-random selection
4. Audio features, as in Concatenative Synthesis [10]

5.1.2 Spatial Read Pointer

In addition to the ability to select a single grain (or groups of grains) in space, we implemented a technique to select a sequence of grains using a "Spatial Read Pointer" (Figure 7). Analogous to the read pointer in classical (temporal) granulation, the spatial read pointer orbits around a specified trajectory, and extracts grains that fall within the path.

Sudden movement transpositions applied to the continuous trajectory of the spatial read pointer results in quasi-sequential selection. Random triggering of the spatiotemporal grains is achieved by providing the algorithm with random values for space, and time.

To ensure that the spatial read pointer is able to extract grains at the correct position in space, the orbit needs to be updated at a rate that is at least as high as the trigger rate. This is achieved by calculating the orbit trajectory in the audio callback, at audio rate. As such, not only are the grains extracted from the correct position in space and time, but the movement of the orbit can be increased to audio rate.

The decoupling of temporal and spatial processes allow us to independently assign the temporal read pointer, as well as the spatial read pointer. The extreme case would be to freeze time, and "scan" the captured space, which would result in spatially exploring a moment frozen in time.

5.1.3 Generative/ Algorithmic

The multidimensional representation allows us to use various algorithms in order to extract, and trigger the spatiotemporal grains.

As we have shown, the spatial read pointer is one technique for specifying a selection pattern. This functions as a starting point for further investigations in extracting, and triggering spatiotemporal grains. Other algorithms that would be explored in the near future include fractal based, physics based, statistical, stochastic, and cellular automaton.

5.2 Spatial Position

Spatial position of grains are now dependent on the encoded spatial information. However, we now have the option of decoupling them so as to reassemble the spatiotemporal grains in a different spatial configuration. For example, one could perform feature analysis of each grain, and spatially (and temporally) group them based on their features, similar to concatenative synthesis [10]

5.3 Spatial Trajectory

Previous methods of assigning spatial trajectory are still applicable. Additionally, spatial trajectory of the grains can be extracted from the sound file, and mapped to a different sound object, or transformed. Examples of transformations include evaporation (sonic disintegration), coalescence (sonic formation) [7], or affine transformations (Section 4.4).

5.4 Grain Density

In classical granulation, when we increase the grain density, we allow more grains to overlap. The contents of the grain (waveform within the grain) can either be the exact copy, i.e. same temporal selection, or at a different time point in the buffer (including optional transformations).

In the realm of spatiotemporal granulation, as each grain contains a different copy of a similar signal (spatial difference), we gain a different effect. The decorrelation of each individual grain allows us to control the source width of a sound object by manipulating spatial grain density.

5.5 Spatial Patterns

In addition to temporal patterns such as synchronous/ asynchronous grain triggering, we now have the ability to construct spatial patterns, derived from the extracted spatial information. These grains can now be synthesized in any arbitrary temporal/ spatial structure.

For example, when we trigger the grains in space via the spatial read pointer, we trigger the grains that fall within the trajectory, allowing us to hear a granular stream extracted from the distinct positions in the encoded space.

6. IMPLEMENTATION: ANGKASA

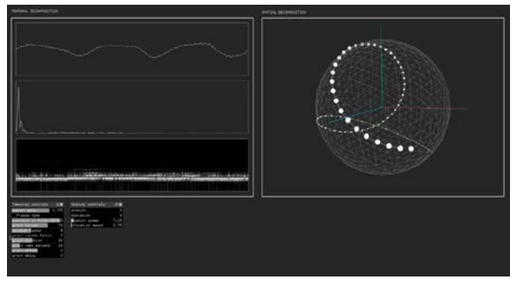


Figure 8: Screenshot of "Angkasa"

The word *Angkasa* originates from the Malay language, derived from the Sanskrit term *Ākāśa*. Although the root word bears various levels of meaning, one of the most common translation refers to "space".

In the context of our research, Angkasa is a software tool which allows a user to analyze, visualize, transform, and perform Spatiotemporal Granulation. The software is designed to be used as a creative tool for composition, real-time musical instrument, or as an analytical tool.

Angkasa was built using openFrameworks (C++ toolkit) on a 2015 Mac Pro (OSX 10.10.5). Documentation of the software can be accessed at www.vimeo.com/157253180.

6.1 User Interface

The Graphical User Interface for Angkasa features a section for temporal decomposition, and a section for spatial decomposition. When used simultaneously, the resulting extraction forms a spatiotemporal grain.

6.1.1 Temporal Decomposition

Visualization of the temporal decomposition includes temporal, and frequency domain plots, as well as a spectrogram to monitor the extracted grains in real time (Figure 8– top left).

Users are able to control parameters such as position in file, freeze time (static temporal window), grain voices, stretch factor, random factor, duration, window type, offset, and delay via GUI sliders.

6.1.2 Spatial Decomposition

The spatial decomposition is visually depicted using a geodesic sphere, which represents the captured sound field. Users specify a value for azimuth & elevation in order to extract grains from that specific position in the captured sound field.

The spatiotemporal grains are visualized as smaller spheres, where the grains are extracted from (Figure 8– top right). The selection of locations can be done via independent GUI sliders, point selector, or algorithmically (discussed in Chapter 5.1).

6.2 Visualization

We plan to improve the visualization so that each grain's characteristics are reflected on its corresponding visual representation (from that location in space).

Furthermore, we plan to map the representation shown in Figures 3 & 4 onto the geodesic sphere shown in Figure 8. This would allow a user to analyze in real-time, before extracting, and triggering the spatiotemporal grains, as well as provide visual queues for the transformations.

7. FUTURE WORK

Development of this research will proceed in different directions, including (but not limited to) analysis, extraction, control, transformation, synthesis, spatialization, and visualization of spatiotemporal grains.

We plan to use Angkasa in the UCSB Allosphere [11], where the spatiotemporal grains can be spatialized via 54 loudspeakers. Additionally, the Allosphere also provides 360° realtime stereographic visualization using a cluster of servers driving 26 high-resolution projectors.

This would allow the spatiotemporal grains to be acoustically, and visually positioned in its corresponding location [12]. An external OSC [13] controller will be designed as a means to navigate the decomposed spatiotemporal palette.

8. CONCLUSION

We presented a novel theory & technique called Spatiotemporal Granulation. This technique uses the inherent spatial encoding from spatially encoded signals, and creates the ability to granulate space and time– resulting in grains that are both spatially, and temporally localized.

Synthesis techniques, and transformations that can be applied to the spatiotemporal grains are outlined, and discussed. The viability of this technique is demonstrated through the software implementation, named "Angkasa".

Acknowledgments

The first author would like to thank his advisors, Curtis Roads, Clarence Barlow, Andres Cabrera, and Matthew Wright for their guidance in this research topic.

This line of research was supported by Universiti Sains Malaysia, Ministry of Education Malaysia, and the Baden-Württemberg Foundation.

9. REFERENCES

[1] C. Roads, *Microsound*. MIT Press, 2001.

[2] B. Truax, "Composition and diffusion: space in sound in space," *Organised Sound*, vol. 3, pp. 141–146, Aug. 1998.

[3] N. Barrett, "Spatio-musical composition strategies," *Org. Sound*, vol. 7, no. 3, pp. 313–323, Dec. 2002.

[4] E. Deleflie and G. Schiemer, "Spatial-grains: Imbuing granular particles with spatial-domain information," in *Proceedings of ACMC09, The Australasian Computer Music Conference*, July 2009.

[5] M. A. Gerzon, "Periphony: With-height sound reproduction," *J. Audio Eng. Soc*, vol. 21, no. 1, pp. 2–10, 1973.

[6] W. Commons. (2013) Spherical harmonics up to degree 3. [Online]. Available: https://commons.wikimedia.org/wiki/Category:Spherical_harmonics#/media/File:Spherical_Harmonics_deg3.png

[7] B. L. Sturm, C. Roads, A. McLeran, and J. J. Shynk, "Analysis, visualization, and transformation of audio signals using dictionary-based methods." in *ICMC*, 2008.

[8] C. Roads, *The Computer Music Tutorial*. MIT Press, 1996.

[9] M. H. W. Rosli and A. Cabrera, "Gestalt principles in multimodal data representation," *Computer Graphics and Applications, IEEE*, vol. 35, no. 2, pp. 80–87, Mar 2015.

[10] D. Schwarz, "A system for data-driven concatenative sound synthesis," in *Digital Audio Effects (DAFx)*, Verona, Italy, 2000.

[11] X. Amatriain, J. Kuchera-Morin, T. Hollerer, and S. T. Pope, "The allosphere: Immersive multimedia for scientific discovery and artistic exploration," *IEEE Multimedia*, vol. 16, no. 2, pp. 64–75, 2009.

[12] M. H. W. Rosli, A. Cabrera, M. Wright, and C. Roads, "Granular model of multidimensional spatial sonification." Maynooth, Ireland: Sound and Music Computing, 2015.

[13] M. Wright and A. Freed, "Open sound control: A new protocol for communicating with sound synthesizers." Thessaloniki, Hellas: International Computer Music Association, 1997.

The Sound Analysis Toolbox (SATB)

Tae Hong Park
Music Technology and Composition
NYU Steinhardt

thp1@nyu.edu

Sumanth Srinivasan
Electrical and Computer Engineering
NYU Tandon

sumanths@nyu.edu

ABSTRACT

Sound analysis software applications have become commonplace for exploring music and audio, and important factors including responsive/fast data visualization, flexible code development capabilities, availability of standard/customizable libraries/modules, and the existence of large community of developers have likewise become integral. The widely used `MATLAB` *software, in particular, has played an important role as an all-purpose audio exploration and research tool. However, its flexibility and practicality when exploring large audio data, its limitations for synchronized audiovisual exploration, and its deficiencies as an integrated system for audio research is an area that can be improved. In this paper we report on developments on the Sound Analysis Toolbox (SATB), a pure* `MATLAB`*-based toolbox that addresses some of* `MATLAB`*'s basic deficiencies as an audio research platform. We introduce solutions including efficient visualization for literally any sized data, a simple feature extraction "plug-in" API, and the sMAT Listener module for spatiotemporal audio-visual exploration.*

1. INTRODUCTION

The *Sound Analysis Toolbox* (SATB) project's origin can be traced to the *EASY Toolbox*. EASY project started as an effort to embrace music information retrieval (MIR) for electro-acoustic music analysis by observing its popularity within the traditional tonal/rhythm research community. EASY included a number of features including implementation of 27 feature extraction algorithms as well as a basic classification module to facilitate the idea of utilizing both qualitative and quantitative approaches for interpreting electro-acoustic music [1]. The research emphasis in exploring electro-acoustic music analysis techniques from a quantitative approach was in part, due to the observation of the genre itself, where an emphasis of non-traditional musical parameters, commonly outside of the realm of melody, pitch structures, harmony, rhythm, and pulse are commonplace. As such, a number of visualizations, including the *timbregram* were developed as shown in Figure 1. The timbregram and other visualizations essentially offered a low-level acoustic descriptor approach for electro-acoustic music exploration and analysis in addition to traditional waveforms and spectrograms. This was primarily accomplished by mapping, and assigning feature clusters to various 3D visualization formats. The goal of EASY was to begin exploring the potential of applying *both* quantitative *and* qualitative analyses paradigms, espousing a more comprehensive music analysis model where both objective *and* subjective approaches for electro-acoustic music analyzing played significant roles [2].

While trying to improve EASY, we to began to recognize a number of design shortcomings, including: (1) *specificity and generality*: from a technical point of view, the EASY Toolbox was narrow in scope in that it was being developed *specifically* for electro-acoustic music. An important design philosophy in EASY was to follow a *timbral* approach to electro-acoustic music analysis, which we thought too rigid in scope; (2) *Analysis Module API*: the EASY Toolbox was implemented using a set of feature analysis modules (feature vector types or classification algorithms) without an API, making third-party contribution, or additional module development cumbersome; and (3) *flexible audio-synchronized visualization*: although many of pre-defined EASY visualizations proved to be insightful, as the EASY Toolbox's visualization tools were not flexible enough to allow customization, we found its utility limiting.

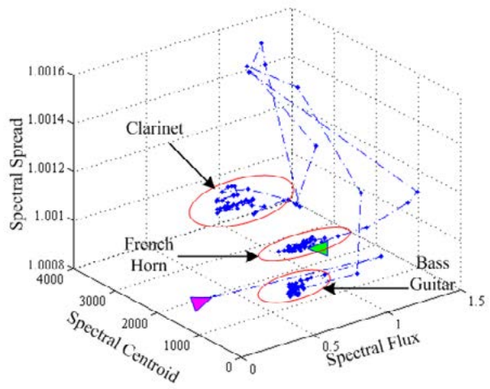


Figure 1. Timbregram: audiovisual exploration of bass, clarinet, and French horn samples

Considering design limitations of the EASY Toolbox, we discontinued its development and began developing SATB [3]. This included broader design philosophies that would facilitate a more general approach to quantitative *music* and *sound* analysis. In particular, as one of our current research is in *Soundscape Information Retrieval* (SIR) [3], we have come to embrace a more modular approach to tool development with the creation of "low-level" analysis tools to facilitate users to customize their own visualizations or analysis algorithms; in short enable users to engage in analysis of all types of music/audio.

In our current version of SATB, we have improved existing EASY modules, added new modules, and created designs that allow for more flexible, customizable, and a `MATLAB`-style interaction platform that we hope will seem familiar to `MATLAB` users and users of other audio research tools. A summary of the SATB system follows a brief survey of sound analysis systems that are currently used today.

1.1 Audio Analysis Tools Examples

There is substantial amount of on-going research and contributions in the field of audio-analysis and music information retrieval (MIR), most of which have approached music analysis from a traditional standpoint – offering analysis outputs such as rhythm analysis, pitch and harmony analysis, and genre classification to name a few. *Sonic Visualiser* [5], for example, provides a wealth of visualizations for audio signals as well as an interface for sound annotation. It also includes a feature extraction plug-in system for customization possibilities. *Wavesurfer* [6] focuses on speech analysis and provides spectrogram visualization while the Python-based *LibROSA* library offers a framework with building blocks to construct MIR systems. *pyAudioAnalysis* [7] is an open source Python library that additionally offers speaker diarization and classification capabilities. While Python is a useful platform for application-centric tools, rapid prototyping and a research-centric approach are still somewhat cumbersome in that a unified research environment is not always available. *MIRToolbox* [8] is `MATLAB` library offers a set of functions for feature extraction such as spectral centroid, tonality, rhythm etc. from audio files, focusing heavily on processing of music in terms of its pitch-duration lattice as opposed to more generic audio signals. The Chroma Toolbox provides implementations for extracting variants of chroma-based features [7] and others are focus on similarity analysis [8]. While all of the aforementioned software is useful and sophisticated in their own ways, they are also fragmented where some lack important yet basic features such as: (1) audiovisual synchronous playback, (2) feature extraction and customizability options, (3) coding environment, and (4) visualization flexibility.

SATB aims to contribute and attempts to consolidate a number of the important fundamental features i.e., fast visualization, general coding platform, feature extraction and classification APIs, while providing a responsive interface in the `MATLAB` environment.

2. SATB

SATB is based on a number of fundamental design philosophies including (1) *familiarity*: the user should find SATB familiar when viewed from the `MATLAB` user-ecosystem; (2) *fast and responsive visualization*: users should be able to quickly "plot" (or `splot` in our case) large data and allow responsive interaction with the data (e.g. zooming, rotation, etc.); (3) *audiovisual synchronization*: the data that is being explored should not only be subject to efficient and quick visualization but also seamless audiovisual exploration so that audio playback is synchronized with "plots" and "subplots"; and (4) *extendible analysis API*: users should be able to use baseline analysis tools such as standard feature extraction algorithms and classification algorithms and also use our APIs to straightforwardly add and contribute custom algorithms as needed. This includes addressing issues concerning customization, contribution to the research community, and easy integration into SATB whereby complexities such as I/O, visualization, and data exploration are handled behind the scenes by the system. These main design components are further summarized in greater detail in the following sub-sections.

2.1 Making a `splot`: Responsive plotting

Large vectors and large files – if they can be loaded into the `MATLAB` workspace at all – are notoriously cumbersome to display and interact with `MATLAB`'s go-to `plot` function. Additionally, although the `MATLAB` `soundplayer` can be used to play audio data (again, if small enough for its workspace), there are no built-in features that provide *synchronous* audiovisual interaction with data. SATB's `splot` addresses shortcomings of these essential features for sound, audio, and music exploration, and furthermore looks and feels the same as `MATLAB`'s `plot` function … but with *added* functionality. SATB's `splot` is essentially a custom, audio-signal-friendly "upgrade" of `MATLAB`'s `plot`. `splot` (or "SATB plot") enables users to quickly display and interact with plots while having access to all of the standard `plot` options such `subplot`, `hold`, `legend`, as well as other `plot` options that `MATLAB` users would expect to be able to use. `splot` utilizes a simple but effective algorithm developed as part of an *iOS DAW* project called *microDAW*[1] and is based on (1) strategically plotting an approximation of a large vector by considering the limited pixels available on digital canvases, (2) strategic re-computation of new estimations of signal portions to be displayed during zoom requests, and (3) exploiting how humans roughly visually perceive large audio signals when displayed with limited resources on computer monitors – i.e. pixels. In essence, the algorithm down-samples the original vector by analyzing windowed portions of the vector that correspond to the computer's canvas pixel width, computing the *min* and *max* values for *each* window, *and* preserving temporal order as shown below where n is the argument and sample index and $x[n]$ is the value at sample index n.

$$\arg\min_{n} x[n] \quad (1)$$

$$\arg\max_{n} x[n] \quad (2)$$

[1] http://www.suitecat.com

SATB internally stores the min-max down-sampled vector which itself is stored in an instance of the SATB's `sFig` (or a "SATB" `figure`) allowing effective memory management (most data types are references via handles to minimize unnecessary resources usage). Zooming into the vector is efficiently implemented by considering *when* to re-compute the requested zoom request of the vector and when to simply *scale* the canvas with the existing down-sampled vector that is already plotted (resampling vs. use of `xlim`). That is, re-compute the "envelope" only when the user requests less than half of the original vectors. We have empirically found that for zoom requests that are larger than 50% of the vector size, the visual difference between a down-sampled vector and the original vector is practically indistinguishable. This allows for each canvas to only plot a maximum of twice the width of the computer's display width in pixels, which makes rotation, zooming in/out or adding sub-plots efficient, effective, and extremely responsive. When zooming into the level at or below the canvas size, down-sampling is bypassed, and requested samples (< canvas width size) are displayed directly as shown in Figure 2.

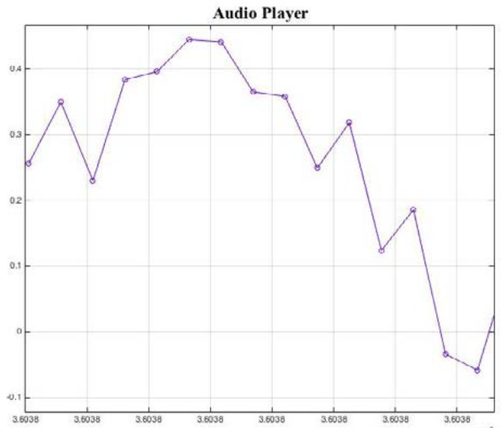


Figure 2. `splot` zoom in canvas level

Zooming out (double-click as commonly done in `MATLAB`) to its original full-vector *overview* is instantaneous as we store the fully zoomed-out and down-sampled vector in a given `MATLAB axes`. Instantaneous zoom-out is equivalent to the size of the initial down-sampled vector: this compressed approximation of the signal under consideration is very compact and stored in the `sFigure`. It is only twice the size of the user's computer monitor pixel width by default. We have found "oversampling" by factor of two worked well for efficient zoom performance (this oversampling factor, however, is customizable).

Using `subplot` and other options such as `hold, line` color, and line style are also seamlessly integrated into `splot` by using `try-catch` statements which bypasses the need for any custom error checking code in SATB – we simply use `plot`'s error checking feature to check syntax and errors for standard `plot` options. `MATLAB`'s `subplots` feature is also integrated into `splot` by using a dynamically changing global down-sampling rates when multiple plots are requested. Here all vectors in a figure's `subplot` are analyzed to compute the global decimation ratio, where the `subplot` with the largest vector size is selected at each `subplot` request. This ensures that all subplots are formatted with same decimation ratio effectively resulting in "apple vs. apple" visualization.

2.1.1 Plotting multidimensional vectors

For multidimensional vectors such as STFT spectrograms, for example, a similar *min-max*, down-sampling algorithm is employed. Instead of down-sampling a one-dimensional "line," in the case of the spectrogram, a rectangular 2D *area* is analyzed for min/max arguments in two dimensions – time and frequency indexes in the case of STFT displays. However, any vector with two dimensions can be plotted and `splot` simply analyzes 1D or 2D data.

2.1.2 Plotting vectors and files

`splot` can handle a number of different data formats. Vectors already in the `MATLAB` workspace can simply be plotted using the exact same syntax used in `plot`. Additionally, `splot` can also display files not in `MATLAB`'s workspace including audio files (all audio file extensions that are recognized in `MATLAB's audioread`), binary files (user will have to provide bit depth and vector dimensional information as a separate `cell` array input argument (e.g. `splot({'fs', 8000})`), or files mapped via `MATLAB`'s `memmapfile` (memory map to a file). The threshold for using SATB's down-sampling feature is customizable and is set to 2 million samples by default.

Title	Music	plot	splot
Beatles	2:21	0.8797	0.5327
Queen	3:36	0.7847	0.6554
Radiohead	6:23	1.2380	0.7852
Coltrane	13:39	3.6341	1.2259
Chowning	17:03	5.4389	1.4146

Table 1. Load time of audio data at various zoom levels

2.1.3 `splot` *benchmarking*

Table 1 shows benchmarking results for `splot` `MATLAB`'s `plot` function. Results for a number of different audio files (at sampling rate 44.1 kHz) such as classic compositions including *Help!* (The Beatles), *Paranoid Android* (Radiohead), *My Favorite Things* (John Coltrane), and *Stria* (John Chowning) are shown. Benchmarking tests were also conducted using sinusoid and Gaussian white noise signals of varied durations to examine performance of our min-max algorithm for decimation. The following figures show performance comparison of `plot` and `splot` functions: duration in seconds required for reading the data, displaying the data, and making `splot`/`plot` responsive to user interaction.

While `plot` is minimally faster (in order of milliseconds) for files with short duration, substantial savings in setup time was observed when using `splot`. Efficiency was observed to be proportional to the number of samples, and consequently, duration of the signal. Figure 3 shows `splot` (solid line) and `plot` performance as signal size is increased in one-minute increments up to 20 minutes: *x* axis is signal duration (min) and *y* axis plotting duration in seconds.

The reader will note that in Figure 3, `splot` is approximately 300% times faster than `plot` for a minute sinusoidal signal. Similar benefits are shown for Gaussian white noise signals. Table 1 shows display performance for different types of musical signals where again, similar advantages can be seen in `splot` performance over `plot` (388% faster for *Stria*).

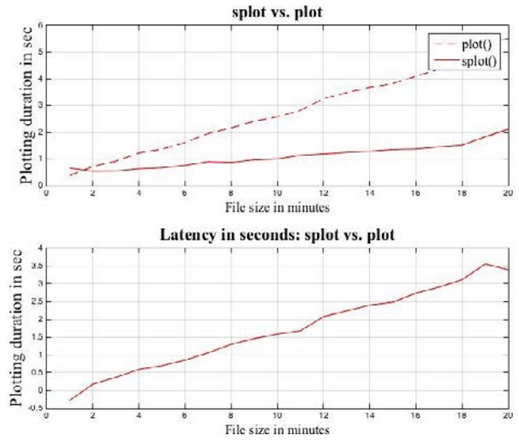


Figure 3. `splot` and `plot` load times for sine signals

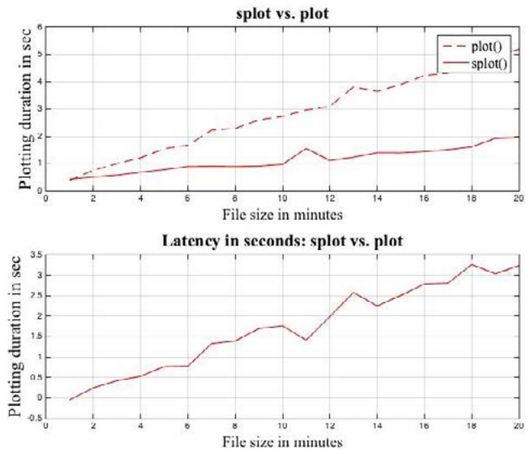


Figure 4. `splot` and `plot` load times for noise

The responsiveness is not only observed when first plotting a signal, but is even more notable when interaction with the plotted data – when zooming, rotating, etc. All benchmarking was done with following hardware and software: MacBook Pro (13-inch, Mid 2012), 2.9G Hz Core Intel i7, 8GB 1600 MHz DDR3, Intel HD Graphics 4000 1024 MB, in MATLAB Version: 8.4.0.150421 (R2015a).

2.1.4 More than splotting: Audiovisual synchronization

The current iteration of `splot` also includes a basic audio transport feature where plotted signals can be played back and synchronized via a dynamically updating cursor to synchronize audio and visuals. This is achieved using `MATLAB DSP System Toolbox` and the `dsp.AudioPlayer`, `step` method, audio queuing, and various customizable latency and audio buffer settings to synchronize audio and visualizations. Current audio transport and audio control features include playback, rewind to start of vector, stop/pause, audio playback sampling rate change, and soloing a *subplot* for playback. Additionally, the SATB interface also provides audio scrubbing, that systems like *Avid Protools* and other DAWs include. Scrubbing is achieved by simply dragging the cursor of a subplot as shown in Figure 5. When the cursor is released, playback resumes at the timestamp corresponding to where mouse button release occurred. For vectors that do not have an associated sampling rate a default value of 44.1 kHz is assigned (sampling rate can be provided as input argument through the `splot` input formatted as `MATLAB cell` array `{…}`).

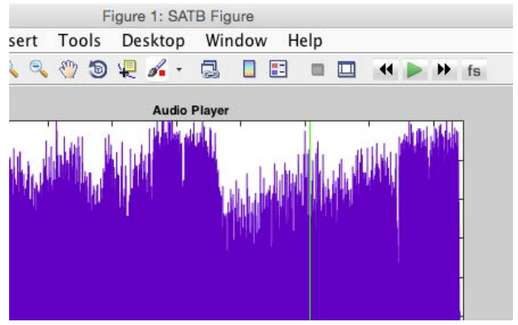


Figure 5. Audio-scrubbing in `SATB`

2.2 Feature Extraction Module

`SATB`'s analysis module currently implements 17 time/frequency-domain low-level feature descriptors including RMS, attack time, crest factor, dynamic tightness [9], low energy ratio, pitch, temporal centroid, zero-crossing rate, MFCC, spectral centroid, spectral flux, spectral jitter, spectral roll-off, spectral shimmer, spectral spread, spectral flatness, and spectral smoothness. SATB's analysis has been designed by considering important factors for audio/music analysis environments, including: (1) *data size flexibility*: analyzing, processing, and storing results, (2) *extendibility and API*: ease of add-

ing additional, custom feature extraction modules, (3) *visualization*: options for adding custom/specialized visualization for any feature extraction module, and (4) *data management*: using handles/references whenever possible to minimize system resources.

Analysis results can be either saved to the `MATLAB` workspace or external storage facilitating large data analysis as well as batch processing. Each feature extraction implementation inherits from an analysis superclass which handles I/O "behind the scenes" as further summarized in *Section 2.1.1*. Each feature extraction module can optionally include a custom visualization method that can be used to display data in specific formats and configurations. Additionally, data management uses `MATLAB` handles/references to help in minimizing duplication of data, easy session organization, and cleanup of `SATB` sessions – this is useful where data management is somewhat lax, especially during data visualization where full resolution data oftentimes exist both in the `MATLAB` workspace a `figure`.

Feature extraction simply begins by creating an `SATB` *instance*, creating a new *session,* and computing features from audio files in external storage devices or vectors in the `MATLAB` workspace. When no options are provided to the SATB constructor, default parameters such as *window size*, *hop size*, *analysis window type*, and *sampling rate* are used for analysis (these default parameters are also user-customizable in the SATB configuration file – the last session's parameters are used). A new session will allow optional creation of a session directory, prompt the user for audio file information, and save all analysis results organized by combining audio file information, analysis types, and feature type. Each session produces an associated SATB `sessionName.mat` file that contains session settings and configurations including dataset information, analysis, pre-processing, and visualization parameters. The SATB feature extraction algorithms can be used on a single vector/audio file or a set of vectors/audio files that can be selected as part of the session's analysis file directory. Other features include bypassing already computed feature vector outputs, selecting feature subsets for analysis, and batch processing of large set of files.

```
function analysis()
  startIdx = 1;
  endIdx   = this.winSize;

  for i=1:this.numOfWin
    this.data.rms(i) = ...
     (mean(this. pcm(startIdx:endIdx).^2))^0.5;

    startIdx = startIdx + this. hopSize;
    endIdx   = endIdx + this.hopSize;
  end
end
```

Figure 6. Simple RMS "plug-in"

2.2.1 Custom feature extraction algorithms and API

Although SATB currently includes a modest 17-feature analysis module, our API allows for easy customization and development of additional feature extraction algorithms. SATB's "plug-in" development architecture is straightforward in that it inherits all necessary methods from its analysis superclass and handles appropriate input/output vector passing to and from each feature extraction module to SATB and the `MATLAB` workspace. Custom third-party contributed feature exaction implementation simply require (1) naming the file as either `td_featureName.m` or `fd_featureName.m`, (2) saving the `.m` file in the `SATB ./features` directory, (3) adding feature dependencies (e.g. `"td_spectralCentroid"`), and finally (4) implementing the feature extraction superclass' `analysis()` method. Everything else is automatically handled by the SATB system, including passing appropriate input vectors to the feature extraction module and saving results. The `analysis()` method for the RMS algorithm is shown in Figure 6.

Additional abstract methods include initialization, preprocessing, and visualization methods to allow customization of the user's feature extraction module. However, for most cases, only the `analysis` abstract method needs to be customized.

Figure 7. sMAT Listener

2.3 sMAT Listener

The *SATB-Matrix Listener* (sMAT) module provides a three-dimensional, audio source matrix-based sound exploration environment where audio stems/tracks are positioned with a 3D virtual space as shown in Figure 7. Each sMAT session can be set up with three general parameters: (1) "stage" image file, (2) "microphone" positions, and (3) initial listening coordinates (or *listening spot*). The image file is used to visually represent a space such as a concert hall "stage" (e.g. Lincoln Center concert hall stage) with a matrix of "microphone" locations (longitude, latitude, and elevation). The microphones are essentially audio files that can be loaded into sMAT where the microphone locations are randomly spread throughout the space when initialized for the first time. The user can then position the "mic nodes" (i.e. soundfiles) within the sMAT 3D space. In the example shown in Figure 7, 22 audio files corresponding to 22 microphone locations are mapped in 3D space. The *listening spot i*s a 3D observation coordinate that can be freely moved around the sMAT space simulating a virtual, "on-stage" listening experience: selecting and moving the listening spot around the 3D space allows, for example, to "eavesdrop" on the string section, percussionist, trumpet player, or experience what the conductor might be hearing on stage, standing on a podium … or what it might sound like facing the audience from the stage rather than the other away around as is more common. All sMAT sessions can be saved and later recalled.

Figure 8. sMAT Listener stage exploration

Figure 8 shows the positioning of the user's listening spot towards the backside of the string section (stage right). Here we note that the listening spot visually highlighting a particular section of the stage/orchestra. Once a session is set up, exploring the space by moving the listening spot, changing perspectives with 3D rotation tool, or zooming in/out of desired locations in the space are some of the ways sMAT can be used for engaging in spatiotemporal sound exploration. The current implementation renders a two-channel audio stream that changes according to the location of the listening spot. The net audio is computed as a function of three-dimensional coordinates of all microphones and panning information.

sMat may be used in numerous situations including exploration of mixing multi track recordings, diffusion multichannel audio playback environments, or exploring soundscapes as is currently being developed as part of our *Citygram* project [10]–[14].

3. FUTURE WORK

We plan to release SATB in the fall of 2016 and much (exciting) work still remains (please refer to `citygram.smusic.nyu.edu` for links/updates to repos) including providing options for "envelope" computation algorithms in addition to our current *min-max*. In particular, for our analysis module, we aim to finish up an API for acoustic event detection (AED) and acoustic event classification (AED). Additionally, we have developed an online sound event annotation module and are in the midst of porting it to `JavaScript` and `WebAudio` for added flexibility. This effort has been developed as part of our soundscape mapping initiatives embracing a multi-listener labeling/annotation philosophy, rather than exclusively relying on one or two researchers' judgments for annotative ground truth. Additionally, we will include database exploration/querying modules for two databases – *Freesound*[2] and *Citygram*[3]. This will allow for easy access to databases including downloading of audio data, labels, and other metadata – directly from `MATLAB`. This feature will be integrated with our sound annotation module.

For sMAT we are currently folding in Park's unpublished software called *soundpath* from 2009 that focuses on spatio-temporal paths as a metaphor for mixing, modulation, chronicling, and annotating event along "sound paths" in the sMAT space. These spatiotemporal soundpaths are played back with other synchronized information, data, and modalities such as historical information, technical details, and musical moments in a composition, soundscape or audio signal. In this context, not only can sMAT be used for exploration in real-time but also in non-real time, especially in education settings where students or instructors can develop narratives to communicate and convey important musical ideas.

Finally, a more long-term sub-module we plan on adding to SATB is feature modulation synthesis (FMS) research [9]. FMS is a feature-centric sound synthesis-by-analysis approach where proof-of-concepts have been implemented in the `MATLAB` environment. The inclusion of FMS in SATB will allow for feature modulation based sound synthesis exploration – e.g. modulating harmonic expansion/compression of stringed instrument sounds which can be used from both creative and research perspectives.

4. CONCLUSIONS

In this paper, we introduced the Sound Analysis Toolbox (SATB) and our currently implemented modules for visualization, sono-visual interaction, and feature extraction. We summarized some of SATB's features including efficient plotting with `splot` taking advantage of computer display limitations, flexible and expandable analysis module and feature extraction APIs, and sMAT as a spatiotemporal sound exploration tool. Our hope is that SATB will contribute in facilitating exploration of music and sound for our community of audio, music, and sound researchers, enthusiasts, musicians, composers, educators, and students alike.

5. REFERENCES

[1] T. Park, D. Hyman, P. Leonard, and W. Wu, "Systematic and quantative electro-acoustic music analysis (sqema)," in *International Computer Music Conference Proceedings (ICMC)*, 2010, pp. 199–206.

[2] T. Park, Z. Li, and W. Wu, "EASY Does it," *… Int. Soc. Music …*, 2009.

[3] T. H. Park, J. Lee, J. You, M.-J. Yoo, and J. Turner, "Towards Soundscape Information Retrieval (SIR)," in *Proceedings of the*

[2] http://www.freesound.org

[3] http://citygram.smusic.nyu.edu

International Computer Music Conference 2014, 2014.

[4] J. Beskow and K. Sjölander, “WaveSurfer-a public domain speech tool,” *Proc. ICSLP 2000*, 2000.

[5] T. Giannakopoulos, “pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis,” *PLoS One*, vol. 10, no. 12, p. e0144610, Dec. 2015.

[6] O. Lartillot and P. Toiviainen, “A Matlab toolbox for musical feature extraction from audio,” *Int. Conf. Digit. Audio …*, 2007.

[7] S. E. Meinard Müller, “Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features.”

[8] E. Pampalk, “A Matlab Toolbox to Compute Music Similarity from Audio.,” *ISMIR*, 2004.

[9] T. Park and Z. Li, “Not just prettier: FMS toolbox marches on,” *Proc. ICMC 2009*, 2009.

[10] T. H. Park, B. Miller, A. Shrestha, S. Lee, J. Turner, and A. Marse, “Citygram One : Visualizing Urban Acoustic Ecology,” in *Proceedings of the Conference on Digital Humanities 2012*, 2012.

[11] C. Shamoon and T. Park, “New York City’s New Noise Code and NYU's Citygram-Sound Project,” in *… -NOISE and NOISE-CON Congress and …*, 2014.

[12] T. H. Park, J. Turner, J. You, J. H. Lee, and M. Musick, “Towards Soundscape Information Retrieval (SIR),” in *International Computer Music Conference Proceedings (ICMC)*, 2014.

[13] T. H. Park, J. Turner, M. Musick, J. H. Lee, C. Jacoby, C. Mydlarz, and J. Salamon, “Sensing Urban Soundscapes,” in *Workshop on Mining Urban Data*, 2014.

[14] C. Mydlarz, S. Nacach, T. Park, and A. Roginska, “The design of urban sound monitoring devices,” *Audio Eng. Soc. …*, 2014.

Short overview in parametric loudspeakers array technology and its implications in spatialization in electronic music

Jaime Reis
INET-md (FCSH-UNL), Festival DME, Portugal
jaimereis.pt@gmail.com

ABSTRACT

In late December of 1962, a Physics Professor from Brown University, Peter J. Westervelt, submitted a paper called Parametric Acoustic Array [1] *considered primary waves interacting within a given volume and calculated the scattered pressure field due to the non-linearities within a small portion of this common volume in the medium* [2]*. Since then, many outputs of this technology were developed and applied in contexts such as military, tomography, sonar technology, artistic installations and others.*

Such technology allows perfect sound directionally and therefore peculiar expressive techniques in electroacoustic music, allowing a very particular music dimension of space. For such reason, it’s here treated as a idiosyncrasy worth to discuss on its on terms.

In 2010-2011 I composed the piece "A Anamnese das Constantes Ocultas", commissioned by Grupo de Música Contemporânea de Lisboa, that used a parametric loudspeakers array developed by engineer Joel Paulo. The same technology was used in the 2015 acousmatic piece “Jeux de l'Espace” for eight loudspeakers and one parametric loudspeaker array.

This paper is organized as follows. A theoretical framework of the parametric loudspeaker array is first introduced, followed by a brief description of the main theoretical aspects of such loudspeakers. Secondly, there is a description of practices that use such technology and their applications. The final section describes how I have used it in my music compositions.

1. Introduction

The fundamental theoretical principles of a parametric loudspeaker array (PLA) were discovered and explained by Westervelt [1]. Interestingly, this was in the same year of the publication of an article by Max Mathews where the author said there were “no theoretical limits to the performance of the computer as a source of musical sounds” [3], a text that then was mentioned by composers who changed the history of computer music, such as John Chowning, as very promising ideas [4], who certainly influenced this and other composers.

A relation between Westervelt discoveries and further developments in parametric loudspeakers array technology were described by Croft and Norris [2], including the technological developments by different scientists and in different countries and how it has moved from theory and experimentation to implementation and application.

It’s important to clear that such terminology isn’t fixed and that it’s possible to find different definitions to similar projects (commercial, scientific or of other nature), uses, products and implementations of such theoretical background, sometimes even by the same authors and in the same articles. Some of them being “parametric loudspeakers” [2], [5], “parametric speakers “ [6], [7], “parametric acoustic array” [1], [8], “parametric array” [5], [7], “parametric audio system” [9], “hypersonic sound” [10], “beam of sound” [1], “audible sound beams” [11], “superdirectional sound beams” [12], “super directional loudspeaker” [13], “focused audio” [14], “audio spotlight” [15], [16], “phased array sound system” [17], among others. The term PLA is being used here since it seems to reunite the main concepts that converge in this technology. Nevertheless, it isn’t meant to be presented as an improved terminology over others. This discussion solely has the purpose of showing that one who might not be familiar with such technology, and wish to research more about it, will find different terms that were originated due to particular historical contexts, manufacturers patents and arbitrary grounds.

2. Theoretical framework

A parametric loudspeaker is guided by a principle described by Westervelt as:

two plane waves of differing frequencies

generate, when traveling in the same direction, two new waves, one of which has a frequency equal to the sum of the original two frequencies and the other equal to the difference frequency [1].

However, to trace a proper theoretical framework of the parametric acoustic array in modern applications, Gan et al. makes a more clear description, based on Westervelt's theory:

When two sinusoidal beams are radiated from an intense ultrasound source, a spectral component at the difference frequency is secondarily generated along the beams due to the nonlinear interaction of the two primary waves. At the same time, spectral components such as a sum-frequency component and harmonics are generated. However, only the difference-frequency component can travel an appreciable distance because sound absorption is generally increased with frequency, and amplitudes of higher-frequency components decay greatly compared with the difference frequency. The secondary source column of the difference frequency (secondary beam) is virtually created in the primary beam and is distributed along a narrow beam, similar to an end-fire array reported in antenna theory. Consequently, the directivity of the difference-frequency wave becomes very narrow. This generation model of the difference frequency is referred to as the parametric acoustic array [8].

The result is that the sound projection from a PLA becomes very narrow, much more than with the use of a regular moving-coil loudspeaker (figure 1).

The dispersion pattern of a loudspeaker may also vary broadly, from omnidirectional to superdirectional. Although it's rare for a speaker to have a truly constant directionality across its entire passband, in part from the fact that most are at least somewhat directional at mid and high frequencies, and, because of the long wavelengths involved, almost unavoidably omnidirectional at low frequencies [18]. Loudspeaker systems exhibit their own radiation patterns, characterized by the technical specification called *dispersion pattern*. The dispersion pattern of a front-projecting loudspeaker indicates the width and height of the region in which the loudspeaker maintains a linear frequency response [19]. Most conventional loudspeakers are broadly directional and one can say they typically project sound forward through a horizontal angle spanning 80 to 90 degrees [12].

Tests in PLA systems have demonstrated angles of circa 15 to 30 degrees at 1 kHz, depending on the used model [20]. Loudspeakers that act as superdirectional sound beams behave like an audio spotlight, focusing sound energy on a narrow spot, typically about 15 degrees in width, making possible that a person can hear a sound, while someone nearby, but outside the beam, does not [12]. Such systems are quite peculiar, even when compared to the so-called *narrow coverage loudspeakers,* that feature dispersion in the 50 degree range, such as some Meyer Speakers [21], and potentially have new applications in many diverse fields.

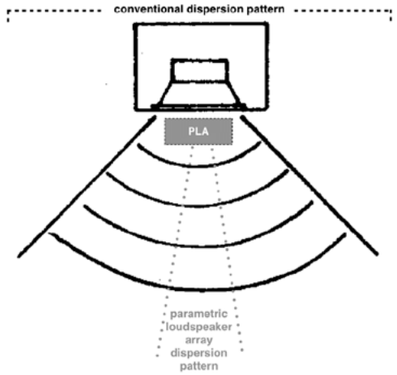


Figure 1. Comparison between hypothetical dispersion patterns for a conventional loudspeaker and for a PLA.

3. Parametric Loudspeakers applications

The proposed applications for such technology vary greatly within the manufacturers of PLA, scientific and artists based proposals, creating a rich interdependence between all fields and hopefully inspiring all involved actors in the creation of new products and synergies. Proposals range from: applications in museum or art galleries, private messaging in vending and dispensing machines, exhibition booths, billboards, multilanguage teleconferencing [5]; acoustic metrology in non destructive testing used on ancient paintings [22]; estimation of acoustical parameters [23]; mobile communication environment creating possibilities for stereo phone calls having a high level of privacy [13]; public safety, security / alarm systems, public speaking [6]; digital signage, hospitals, libraries [15]; control room, tradeshows [14]; automotive applications, slot machines, mobile applications [24]; underwater acoustics, measurement of environmental parameters, sub-bottom and seismic profiling and other naval appliances [25]; and many others, some of them to be further discussed.

While many of the applications use self built devices, there are commercial products that sell PLA, namely, ® Soundlazer [6], ® Holosonics [15], ® Brown Innovations [14], ® Acouspade (by Ultrasonic Audio) [24], ® Hypersonic Sound (LRAD corporation) [26], and others.

Defining the application of PLA in *artistic fields* or within musical practices isn't obvious. In that sense, Blacking clears that "no musical style has 'its own terms': its terms are the terms of its society and culture, and of the bodies of the human beings who listen to it, and create and perform it" [27]. In such terms, is Hiroshi Mizoguchi human-machine interface (named 'Invisible Messenger'), that integrates real time visual tracking of face and sound beam forming by speaker array [28], an *art work*? For the purpose of the present paper, Mizoguchi's work will not be considered as an *art form*, since the authors don't consider themselves as doing *art*. As Bourdieu mentions, one may view the 'eye' as being a product of history reproduced by education, being true for the mode of "artistic perception now accepted as legitimate, that is, the aesthetic disposition, the capacity to consider in and for themselves, as form rather than function, not only the work designated for such apprehension, i.e. legitimate works of art, but everything in the world, including cultural objects which are not yet consecrated" [29].

For the purpose of the present paper, the perspective of the creators will be the base to integrate the use of PLA technology as an application in their *artistic expression* or as other form of expressive behavior. The importance of clearing such categorization is not to imply any form of hierarchy, but merely to formulate a context in presentation, order and grouping of the presented and discussed works.

Other forms of applications are explicitly affirmed as *art practices*, such as the case of Yoichi Ochiai's experiments with ultrasonic levitation [30], that presents himself as a *media artist* [31]. The use of PLA may also be seen in *installations* such as Misawa's "Reverence in Ravine" [32], or "Guilt", by Gary Hill [33]–[35], and reported in *sound art* and *music* by artists such as Miha Ciglar, head of IRZU – Institute for Sonic Arts Research, Ljubljana, Slovenia and creator of several devices and works using PLA, namely a "hands free" instrument, utilizing a non-contact tactile feedback method based on airborne ultrasound or acoustic radiation pressure waves as a force feedback method [36], [37]. Darren Copland has used PLA technology extensively, having created pieces and developed spatialization techniques specific for these, placing a PLA system by Holosonics company in a metal frame with handles on the sides and a mounting point at the bottom that allows the speaker to be rotated 360 degrees on a tripod stand or using a wood frame with handles on the side which are connected with a strap that goes around the performer's neck like the strap for an accordion player [38]. Other artists that have been using such technology relate to the DXARTS - Seattle Arts and Technology, such as Michael McCrea *Acoustic Scan* [39] and Juan Pampin that have present in 2007, with other colleagues, works that were using PLA technology as *ultrasonic waveguides*, as an *acoustic mirror* and as *wearable sound* [40]. Furthermore, Pampin has used PLA technology in musical pieces such as 2014 "Respiración Artificial", for bandoneon, string quartet, and electronics using PLA, as he mentions in an interview:

The piece is about breathing cycles. The bandoneon has a big bellow and is able to hold a note for a very long time. The timing of the inhale and exhale of the instrument was used to define the time structure of the piece. The beginning of my piece is all in the very upper register (above 1000 Hertz, around the C above treble clef). When you hear up there, you hear in a different way. Your ear is not able to resolve what is happening with pitch, the notes tend to shimmer, it builds sensation. This piece is all sensorial it's not theoretical. Its more neurological if you want. In terms of the electronics, I am using a 3D audio system and ultrasonic speakers that we developed in DXARTS. These speakers can produce highly localized beams of sound – akin to spotlights – which can move around the audience and bounce off the architecture of the room [41]

Despite the motivations to interfere in *space* in peculiar ways, that can be read in many of the mentioned articles and websites, the use of PLA in artistic practices hasn't been studied as something particular, possibly, because: it's too recent; such creations operate at individual levels or even when within institutions, they appear to occur locally; or simply because there may be no particular feature that makes worth of distinction by musicologists, art historians, anthropologists or other scientists in the field of social sciences. There are many other applications of PLA being developed at this very moment. The ones presented here represent only a short research about such topic and are not expected to cover the full length of the use of such technology.

Independently of using PLA technology or not, the idea of directing sound in precise ways or, one could say, the idea of working with space as a parameter in sound creation, has been a very important concept in electroacoustic music. Curtis Roads has referred to superdirectional sound beams and their developments, focusing on audio technology and on electroacoustic music [12], [42]. Among other technologies, the author emphasizes the specificity of PLA technology, explaining the involved principles of acoustic heterodyning, first observed by Helmholtz.

When two sound sources are positioned relatively closely together and are of sufficiently high amplitude, two new tones appear: one lower than either of the two original ones and a second one that is higher than the original two. The two new combination tones correspond to the sum and the difference of the two original ones. For example, if one were to emit two ultrasonic frequencies, 90 KHz and 91 Khz, into the air with sufficient energy, one would produce the sum (181 kHz) and the difference (1 kHz), the latter of which is in the range on human hearing. Helmholtz argued that the phenomenon had to result from a non linearity of air molecules, which begin to behave nonlinearly (to heterodyne or intermodulate) at high amplitudes [12].

The author continues detailing that the main difference between regular loudspeakers and loudspeakers that use acoustical heterodyning (PLA), is that they project energy in a collimated sound beam, making an analogy to the beam of light from a flash-light and giving the example that one can direct the ultrasonic emitter toward a wall and a listener in the reflected beam perceives the sound as coming from that spot. Mentioning that, however, "at the time of this writing, there has been little experimentation with such loudspeakers in the context of electronic music" [12].

4. Parametric Loudspeakers in my music

In 2010-2011, I composed the piece "A Anamnese das Constantes Ocultas", commissioned by and dedicated to Grupo de Música Contemporânea de Lisboa (GMCL). The piece was conceived for nine players - soprano voice, flute, clarinet, percussion, harp, piano, violin, viola, violoncello; with conductor and electronics: six regular loudspeakers, one directional PLA loudspeaker, amplified hi-hat, using a click track for the conductor (figure 2).

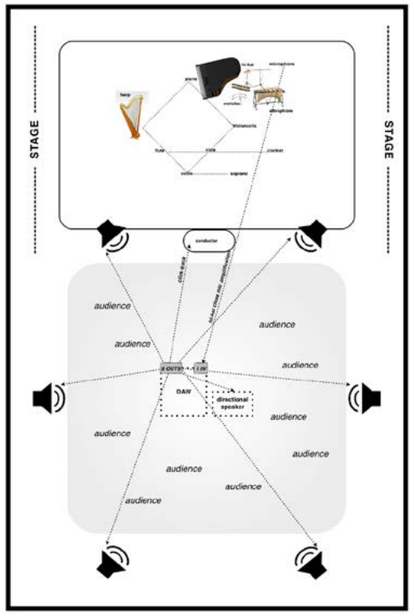


Figure 2. Schema for the disposition of loudspeakers and instruments for the performance of "A Anamnese das Constantes Ocultas".

The players are to be set on stage and the electronic diffused in the six conventional loudspeakers, to be distributed around the audience. The PLA requires an operator to play it. The score has specific instructions demonstrating at each moment where to point (what kind of surfaces to point at, or "swipe" the complete audience or just parts of the audience). One extra musician is required to operate the electronics, in order to control the amplitude of the fixed media electronics (both for the regular loudspeakers and the PLA), the hi-hat amplification and the players amplification (when necessary).

The experimentation and development of the piece was only possible by the dedication of GMCL and engineer Joel Paulo, who developed a parametric loudspeakers array for this piece. At the beginning of the composition I had only heard about such technology, but had never tested it.

Figure 3. GMCL playing "A Anamnese das Constantes Ocultas"; Salão Nobre of Escola de Música do Conservatório Nacional (Lisbon); 26th May 2012; Musicians: Susana Teixeira (voice), Cândido Fernandes (piano), João Pereira Coutinho (flute), José Machado (violin), Luís Gomes (clarinet), Ricardo Mateus (viola), Fátima Pinto (percussion), Jorge Sá Machado (cello), Ana Castanhito (harp); conductor: Pedro Neves. Photo: Cristina Costa.

Figure 4. Rehearsals in the same concert. PLA operator: Joana Guerra; electronics: Jaime Reis.

In this piece, the electronics have three fundamental grounds:

1) generate *large architectural spaces* through the hi-hat amplification, using very close miking (less than 1 cm, using a condenser microphone) of the hi-hat, combined with timbre transformations and spatialization of the signal trough the six regular loudspeakers and the PLA; with such close miking, there are significant changes in the hi-hat timbre, in order to create the idea of playing a huge non pitched gong that it should sound as if it was in a big *pyramid*; different areas of the spectra are distributed in space using both the regular loudspeakers (generally using low and mid frequency, whose range usually changes gradually) and the PLA (dedicated to higher frequencies and distributed in the room in reflective surfaces such as walls, ceiling and floor); resonators were also applied to such timbres that have common pitches with the instrumental textures;

2) *new dimensions* in instrumental spatialization, using the PLA as an extension of instrumental melodic lines, besides punctual diffusion in the regular loudspeakers; the combined use of a timbre and pitch in acoustic instruments, regular loudspeakers and PLA generates very peculiar perceptions of location and source identification;

3) a semantic approach to unveil *hidden messages* that are sung live and in the electronics (mainly in the PLA); the poem to be sung is polysemic and its different meanings are suggested in the prosody, mainly differentiated in this piece by rhythm; the use of the so-called *hidden messages* appear as a reinforcement for the intended meaning, punctually completely revealed by the singers in spoken text; due to the high degree of directivity, such passages should be pointed directly at the audience, making it possible that only specific parts of the audience will listen to those exact passages; more than the usual problem of a member of the audience staying outside of the *sweet spot* and not being able to listen to the spatialization in the same way (as often occurs in acousmatic music), here the purpose is to make each performance unique and somehow personalized, in the sense that the PLA operator may direct sound just for one person or a group of people (that I call *direct operations*); this is different from the *reflective operations* of the PLA (in both figures 5 and 6), constituted by the moments when the PLA is pointed at a surface and the sounds are diffused in the room.

The use of the PLA was integrated from the beginning in the piece's structure and it isn't possible to play the piece without such technology.

Other piece that requires PLA is the 2015 acousmatic work "Jeux de l'Espace", for eight regular loudspeakers, equidistant around the audience (such as in a regular octophonic system) and one directional PLA loudspeaker (to be operated during performance either in the center of the octophony or in front of the audience).

Figure 5. Premiere of the piece "Jeux de l'Espace" in Festival Monaco Électroacoustique, 30th May 2015; playing the PLA in the center of Théâtre des Variétés, using Michel Pascal's (on the left) and Gaël Navard's *Acousmonium* du CNRR de *Nice. Photo:* [43].

Although the PLA movements also have to be precise for each moment of the composition (requiring adaptations to the performance architectural space), there were other principles involved for this composition. It was inspired in *space* as a musical parameter and as the cosmos, integrating sounds derived from processes of sonification from NASA and ESA. The intention is to create an imaginary of a *cosmic momentum* were space is experienced in a tridimensional octophonic sound system with an additional spatial dimension of sound created by the PLA.

In this piece, the main principles of working *space* as a musical parameter are:

1) working on the limits of perception of spatial movements, for example, varying the speed of rotations, based on my own perception of what is heard as a rotation or, if too fast, as a texture of points whose movements in space cannot be perceived in their directionality;

2) create spatial movements that are similar, meaning, identifiable as being connected, like identical paths, or in opposite direction, or symmetric; where the used sounds change in envelope, timbre, rhythm and pitch in order to make such paths more or less identifiable, such as in a gradual scale of levels of identification that is used to make such paths more clear in some situations than in others;

3) compose moments of *hybrid* spatialization using the octophonic system and the PLA in *indistinguishable* ways were the fusion between PLA sound and the regular loudspeakers sound doesn't allow a precise perception of the sound source; this is usually achieved by using *reflective operations* of the PLA simultaneously with the use of the octophonic system as an extension of the PLA (or PLA as an extension of the octophony), and connecting both in timbre and *gestures*;

4) compose moments of *independent* spatialization of both systems; such as PLA *solos* than can be arranged in different ways with different degrees of elucidating the listener to the on going spatial processes: to play the PLA as a *soloist* (as if it was an instrument playing with an orchestra) with *direct operations* while operating the octophony in a more detached way from the PLA; use PLA *solos* (without octophony) using mainly *reflective operations* and punctually *direct operations.*

Figure 6. Performance of the piece "Jeux de l'Espace" in Santa Cruz airfield (Aeroclube de Torres Vedras); 25th June 2015; schema exemplifying *reflective operations*, in this case, pointing the PLA to the floor. Photo: [44]

I have only read about the works and research of Darren Copeland, Miha Ciglar, Juan Pampin and others recently, years after starting using such technology. Even so, it was interesting to note that many aspects that I have mentioned are similar to other composers, namely some of the spatialization techniques used by Copland [38]. Other elements about the construction of sounds, form and other compositional elements could be discussed, but will be left for further discussions and at the light of new research in this field.

5. Conclusions

By the its name, the concept of a "4th dimension" could be expressed in the sound system *4DSound* [45], [46]. The creators of this system decided to refer to the idea of a fourth dimensional sound not by using *superdirectional sound beams*, but using omnidirectional loudspeakers, with experiments in different fields, one of the most significant being from one of it's designers, Paul Oomen, in his opera "Nikola" [47].

However, the title of this presentation wasn't taken from *4DSound*, but from a reflection based on my experience with PLA technology. To answer to the proper application of the concept of a fourth dimension in (electronic) music isn't simple. In modern physics, space and time are unified in a four-dimensional Minkowski continuum called *spacetime*, whose metric treats the time dimension differently from the three spatial dimensions. Since a fourth dimension is considered the *spacetime continuum* and sound waves exist *within* it in the way such *continuum* has in itself three dimensional material space which is where sound waves exist, one could argue if such dimension could exist in sound by questioning *where* and *when* such dimension could be found.

Considering this, one could question if PLA can be considered a fourth dimension of space in electronic music? I would have to answer: no, because I don't think the concept of a fourth dimension is applied to sound simply by using PLA technology. However, I do believe that such use implies indeed a *new dimension* in space and in our perception of it, making a new parameter to consider while composing or working with sound. And, if not, one could ask why even to consider such concept as a main question. The answer to that is merely empirical, because in the last five years that I have worked with PLA technology and presented it in my tours in Europe, America and Asia, this question would very often come from people in audiences of concerts and conferences: *is it like a fourth dimension of sound?* So, it seemed to be a good question to reflect.

The use of PLA is in expansion in many fields. The novelty doesn't appear to be in the technology itself (since it's around for decades), but on the way it's being used. The *how* and *whys* for each creator or group of creators are to be intensively developed and studied.

6. References

[1] P. J. Westervelt, "Parametric Acoustic Array," *J. Acoust. Soc. Am.*, vol. 35, no. 4, pp. 535–537, 1963.

[2] J. J. Croft and J. O. Norris, "Theory, History, and the Advancement of Parametric Loudspeakers: A TECHNOLOGY OVERVIEW Rev. E," *HYPERSONIC SOUND - Am. Technol. Corp.*, 2003.

[3] M. V Mathews, "The Digital Computer as a Musical Instrument," *Science (80-.).*, vol. 142, no. 3592, pp. 553–557, 1963.

[4] J. M. Chowning, "Digital sound synthesis, acoustics and perception: A rich intersection," in *COST G-6 Conference on Digital Audio Effects*, 2000, pp. 1–6.

[5] C. Shi and W.-S. Gan, "DEVELOPMENT OF A PARAMETRIC LOUDSPEAKER: A NOVEL DIRECTIONAL SOUND GENERATION TECHNOLOGY," *IEEE Potentials*, vol. NOVEMBER/D, pp. 20–24, 2010.

[6] "SoundLazer." .

[7] F. J. Pompei, "Ultrasonic transducer for parametric array." Google Patents, 2013.

[8] W.-S. Gan, J. Yang, and T. Kamakura, "A review of parametric acoustic array in air," *Appl. Acoust.*, vol. 73, no. 12, pp. 1211–1219, Dec. 2012.

[9] F. J. Pompei, "Parametric audio system." Google Patents, 2011.

[10] W. Norris, "Hypersonic sound and other inventions." TEDTalk, 2004.

[11] F. J. Pompei, "The Use of Airborne Ultrasonics for Generating Audible Sound Beams," *J. Audio Eng. Soc*, vol. 47, no. 9, pp. 726–731, 1999.

[12] C. Roads, *Composing Electronic Music: A New Aesthetic*. New York: Oxford University Press, 2015.

[13] Y. Nakashima, T. Yoshimura, N. Naka, and T. Ohya, "Prototype of Mobile Super Directional Loudspeaker," *NTT DoCoMo Tech. J.*, vol. 8, no. 1, pp. 25–32, 2006.

[14] "BrownInnovations." .

[15] "Holosonics." .

[16] M. Yoneyama, J. Fujimot, Y. Kawamo, and S. Sasabe, "The audio spotlight: An application of nonlinear interaction of sound waves to a new type of loudspeaker design," *J Acoust Soc Am*, vol. 73, no. 5, pp. 1532–1536, 1983.

[17] J. Milsap, "Phased array sound system." Google Patents, 2003.

[18] T. Rossing, Ed., *Springer Handbook of Acoustics*, 2nd ed. New York: Springer-Verlag, 2014.

[19] C. Roads, *The Computer Music Tutorial*. Massachusetts: The MIT Press, 1996.

[20] F. Pokorny and F. Graf, "Akustische Vermessung parametrischer Lautsprecherarrays im Kontext der Transauraltechnik," in *40. Jahrestagung der Deutschen Gesellschaft für Akustik*, 2014.

[21] "UPQ-2P : Narrow Coverage Loudspeaker," *Meyer Sound Laboratories*. 2008.

[22] S. De Simone, L. Di Marcoberardino, P. Calicchia, and J. Marchal, "Characterization of a parametric loudspeaker and its application in NDT," in *Acoustics 2012*, 2012.

[23] J. V. C. P. Paulo, "New techniques for estimation of acoustical parameters,"

Universidade Técnica de Lisboa, 2012.

[24] "Ultrasonic-audio." .

[25] A. O. Akar, "CHARACTERISTICS AND USE OF A NONLINEAR END-FIRED ARRAY FOR ACOUSTICS IN AIR," NAVAL POSTGRADUATE SCHOOL, 2007.

[26] "Hypersonic Sound." .

[27] J. Blacking, *How Musical is Man?*, 6th Editio. USA: University of Washington Press, 2000.

[28] H. Mizoguchi, Y. Tamai, K. Shinoda, S. Kagami, and K. Nagashima, "Visually steerable sound beam forming system based on face tracking and speaker array," in *IPCR - Conference on Pattern Recognition*, 2004.

[29] P. Bourdieu, "Distinction & The Aristocracy of Culture," in *Cultural Theory and Popular Culture: A Reader*, J. Storey, Ed. Athens: The University of Georgia Press, 1998, pp. 431–441.

[30] Y. Ochiai, T. Hoshi, and J. Rekimoto, "Three-dimensional Mid-air Acoustic Manipulation by Ultrasonic Phased Arrays," *PLoS One*, vol. 9, no. 5, 2014.

[31] "Yoichi Ochiai Youtube Channel." .

[32] D. Misawa, "installation: 'Reverence in Ravine.'" 2011.

[33] R. C. Morgan, "Gary Hill," 2007. .

[34] L. M. Somers-Davis, "Postmodern Narrative in Contemporary Installation Art." .

[35] G. Hill, "Guilt - media installation." 2006.

[36] M. Ciglar, "An ultrasound based instrument generating audible and tactile sound," in *Conference on New Interfaces for Musical Expression (NIME)*, 2010, pp. 19–22.

[37] M. Ciglar, "Tactile feedback based on acoustic pressure waves," in *ICMC - International Computer Music Conference*, 2010.

[38] D. Copeland, "The Audio Spotlight in Electroacoustic Performance Spatialization," in *eContact! 14.4 — TES 2011: Toronto Electroacoustic Symposium / Symposium Électroacoustique de Toronto*, 2011.

[39] M. McCrea and T. Rice, "Acoustic Scan," 2014.

[40] J. Pampin, J. S. Kollin, and E. Kang, "APPLICATIONS OF ULTRASONIC SOUND BEAMS IN PERFORMANCE AND SOUND ART," in *International Computer Music Conference (ICMC)*, 2007, pp. 492–495.

[41] S. Myklebust, R. Karpan, and J. Pampin, "Musical Experimentation Happens on Campus with the JACK." 2014.

[42] C. Roads, *Microsound*. Cambridge, MA: MIT Press, 2004.

[43] "Jaime Reis - Personal Website." .

[44] "Festival DME - Dias de Música Electroacústica Website." .

[45] T. Hayes, "How The Fourth Dimension Of Sound Is Being Used For Live Concerts," *FastCoLabs*, Dec-2012.

[46] J. Connell, F. To, and P. Oomen, "4DSOUND." .

[47] P. Oomen, S. Minailo, K. Lada, R. van Gogh, Sonostruct~, One/One, M. Warmerdam, K. Walton, S. Breed, VOI-Z, M. de Roo, and E@RPORT, "Documentary: Nikola Technopera." 2013.

Extended Convolution Techniques for Cross-Synthesis

Chris Donahue
UC San Diego
cdonahue@ucsd.edu

Tom Erbe
UC San Diego
tre@ucsd.edu

Miller Puckette
UC San Diego
msp@ucsd.edu

ABSTRACT

Cross-synthesis, a family of techniques for blending the timbral characteristics of two sounds, is an alluring musical idea. Discrete convolution is perhaps the most generalized technique for performing cross-synthesis without assumptions about the input spectra. When using convolution for cross-synthesis, one of the two sounds is interpreted as a finite impulse response filter and applied to the other. While the resultant hybrid sound bears some sonic resemblance to the inputs, the process is inflexible and gives the musician no control over the outcome. We introduce novel extensions to the discrete convolution operation to give musicians more control over the process. We also analyze the implications of discrete convolution and our extensions on acoustic features using a curated dataset of heterogeneous sounds.

1. INTRODUCTION

Discrete convolution (referred to hereafter as *convolution* and represented by $*$) is the process by which a discrete signal f is subjected to a finite impulse response (FIR) filter g to produce a new signal $f * g$. If f has a domain of $[0, N)$ and is 0 otherwise and g has a domain of $[0, M)$, then $f * g$ has a domain of $[0, N + M - 1)$. We define convolution as Eq. (1).

$$(f * g)[n] = \sum_{m=0}^{M-1} f[n-m]\, g[m] \tag{1}$$

The convolution theorem states that the Fourier transform of the result of convolution is equal to the point-wise multiplication of the Fourier transforms of the sources. Let $\mathcal{F}$ denote the discrete Fourier transform operator and $\cdot$ represent point-wise multiplication. An equivalent definition for convolution employing this theorem is stated in Eq. (2) and is often referred to as *fast convolution*.

$$\begin{aligned} \mathcal{F}(f * g) &= \mathcal{F}(f) \cdot \mathcal{F}(g) \\ &= \|\mathcal{F}(f * g)\|\ e^{\mathrm{i}\,\angle\mathcal{F}(f*g)} \\ \text{where } \|\mathcal{F}(f * g)\| &= \|\mathcal{F}(f)\| \cdot \|\mathcal{F}(g)\|, \\ \angle\mathcal{F}(f * g) &= \angle\mathcal{F}(f) + \angle\mathcal{F}(g) \end{aligned} \tag{2}$$

When employing convolution for cross-synthesis, one of the two sounds is interpreted as an FIR filter and applied to the other. There are several issues with convolutional cross-synthesis that restrain its musical usefulness.

Treating one of the sounds as an FIR filter essentially interprets it as a *generalized resonator* [1]. However, because convolution is a commutative operation the process is akin to coupling two resonators. The results of convolutional cross-synthesis are often consequently unpredictable and ambiguous. Additionally, there is no way to skew the influence over the hybrid result more towards one source or the other.

Another issue with convolutional cross-synthesis is that the frequency spectra of naturally-produced sounds is likely to decrease in amplitude as frequency increases [2]. The convolution of two such sounds will result in strong attenuation of high frequencies which has the perceived effect of diminishing the "brightness" of the result.

Early attempts to remedy the brightness issue, especially with regard to the cross-synthesis of voice with other sounds (vocoding), involved a preprocessing procedure. A "carrier" sound would be whitened to bring its spectral components up to a uniform level to more effectively impress the spectral envelope of a "modulator" sound onto it [3]. While effective at increasing the intelligibility of the modulator, preprocessing still leaves the musician with limited control over the cross-synthesis procedure and is not particularly generalized.

We introduce an extended form of convolution for the purpose of cross-synthesis represented by $\hat{*}$. This formulation allows a musician to navigate a parameter space where both the perceived brightness of the result as well as the amount of influence of each source can be manipulated. We define extended convolution in its full form as Eq. (3). We will present our justification of these extensions from the ground up in Section 2 and analyze their effect on acoustic features in Section 3.

$$\begin{aligned} \mathcal{F}(f \hat{*} g) &= \|\mathcal{F}(f \hat{*} g)\|\ e^{\mathrm{i}\,\angle\mathcal{F}(f\,\hat{*}\,g)} \\ \text{where } \|\mathcal{F}(f \hat{*} g)\| &= (\|\mathcal{F}(f)\|^p \cdot \|\mathcal{F}(g)\|^{1-p})^{2\,q}, \\ \angle\mathcal{F}(f \hat{*} g) &= 2\,s\,(r\,\angle\mathcal{F}(f) + (1-r)\,\angle\mathcal{F}(g)) \end{aligned} \tag{3}$$

2. EXTENDING CONVOLUTION

In this section we will expand on our extensions to convolution based on the two criteria we have identified: control over the brightness and source influence over the outcome.

2.1 Brightness

Convolution of arbitrary sounds has a tendency to exaggerate low frequencies and understate high frequencies. One way to interpret the cause of this phenomenon is that the

magnitude spectra of the two sounds constructively and destructively interfere with each other when multiplied during convolution. The interference of low-frequency peaks in natural sounds is likely to yield higher resultant amplitudes than the interference of high-frequency peaks.

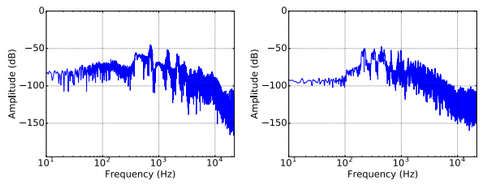
To resolve this issue, we employ the geometric mean when combining the magnitude spectra of two sounds. The geometric mean mitigates both the constructive and destructive effects of interference resulting in a more flattened spectrum. In Eq. (4), we alter the form of the convolved magnitude spectrum from Eq. (2).

$$\|\mathcal{F}(f \hat{*} g)\| = \sqrt{\|\mathcal{F}(f)\| \cdot \|\mathcal{F}(g)\|} \tag{4}$$

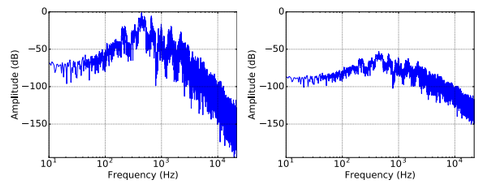
More generally, we introduce a parameter q that controls the flatness of the hybrid magnitude spectrum in Eq. (5).

$$\|\mathcal{F}(f \hat{*} g)\| = (\|\mathcal{F}(f)\| \cdot \|\mathcal{F}(g)\|)^{q} \tag{5}$$

Note that this formulation collapses to *ordinary convolution* as defined in Eq. (2) when $q = 1$ and *geometric mean magnitude convolution* as defined in Eq. (4) when $q = 1/2$. As q decreases towards 0, the magnitude spectrum flattens resulting in noisier sounds. We demonstrate this effect in Figure 1. As q increases past 1, constructive interference between the frequency spectra of f and g is further emphasized, eventually resulting in tone-like sounds.



(a) Source 1 magnitude spectrum (b) Source 2 magnitude spectrum



(c) Ordinary convolution magnitude spectrum ($q = 1$) (d) Geometric mean magnitude convolution spectrum ($q = 1/2$)

Figure 1: Example of cross-synthesis of two sounds (Figure 1a and Figure 1b) using ordinary convolution (Figure 1c) and geometric mean magnitude convolution (Figure 1d).

Eq. (6) is an alternative but equivalent method of calculating $\mathcal{F}(f \hat{*} g)$ with magnitude as defined in Eq. (5) and phase as defined in Eq. (2). It does not use any trigonometric functions and can generally be computed faster in conventional programming environments.

$$\mathcal{F}(f \hat{*} g) = \frac{(a\,c - b\,d) + \mathrm{i}\,(b\,c + a\,d)}{((a^2 + b^2)\,(c^2 + d^2))^{\frac{1-q}{2}}}, \tag{6}$$
$$\text{where } a = \Re(\mathcal{F}(f)),\ b = \Im(\mathcal{F}(f)),$$
$$c = \Re(\mathcal{F}(g)),\ d = \Im(\mathcal{F}(g))$$

2.2 Source Emphasis

We would like the ability to "skew" emphasis of the cross-synthesis result more towards one sound or the other. Our separation of sources into magnitude and phase spectra via fast convolution allows us to modify the amount of influence each source has over the result.

2.2.1 *Skewed Magnitude*

We extend Eq. (5) to Eq. (7), adding a parameter p which allows the influence of source magnitude spectra $\|\mathcal{F}(f)\|$ and $\|\mathcal{F}(g)\|$ to be skewed in the outcome $\|\mathcal{F}(f \hat{*} g)\|$.

$$\|\mathcal{F}(f \hat{*} g)\| = (\|\mathcal{F}(f)\|^{p} \cdot \|\mathcal{F}(g)\|^{(1-p)})^{2\,q} \tag{7}$$

With this form $p = 1$ fully emphasizes $\|\mathcal{F}(f)\|$, $p = 0$ fully emphasizes $\|\mathcal{F}(g)\|$, and $p = 1/2$ emphasizes neither. As p skews further towards 0 or 1, one of the source's magnitude spectrum is increasingly flattened and the result becomes akin to vocoding. We multiply q by the coefficient 2 to maintain the same scale as in Eq. (5) when $p = 1/2$.

2.2.2 *Skewed Phase*

We make a similar extension for the phase of the outcome in Eq. (8), adding a parameter r which allows the influence of source phase spectra $\angle\mathcal{F}(f)$ and $\angle\mathcal{F}(g)$ to be skewed in the outcome $\angle\mathcal{F}(f \hat{*} g)$.

$$\angle\mathcal{F}(f \hat{*} g) = 2\,(r\,\angle\mathcal{F}(f) + (1 - r)\,\angle\mathcal{F}(g)) \tag{8}$$

With this form $r = 1$ fully emphasizes $\angle\mathcal{F}(f)$, $r = 0$ fully emphasizes $\angle\mathcal{F}(g)$, and $r = 1/2$ emphasizes neither. We multiply by the coefficient 2 to maintain the analogy of parameter r to parameter p. With the addition of r, the original input sounds can be recovered in the extended convolution parameter space ($p = r = [0, 1]$, $q = 1/2$).

2.3 Phase Scattering

We suggest one final extension to convolution for the purpose of cross-synthesis that does not directly address our two core issues of brightness and source influence. Analogous to parameter q for manipulating the hybrid magnitude spectra, we introduce a parameter s to our definition of hybrid phase spectra in Eq. (9).

$$\angle\mathcal{F}(f \hat{*} g) = 2\,s\,(r\,\angle\mathcal{F}(f) + (1 - r)\,\angle\mathcal{F}(g)) \tag{9}$$

As s decreases towards 0, the source phase is nullified resulting in significant amounts of time domain cancellation yielding impulse-like outcomes. As s increases past 1, the source phase is increasingly scattered around the unit circle eventually converging to a uniform distribution. Randomizing phase in this manner is similar to the additive phase noise of [4] and produces ambient-sounding results with little variation in time.

3. ANALYSIS

In this section we detail our analysis of the effects of convolution and our extended convolution techniques on a curated collection of "random" sounds.

3.1 Data Collection

We used the Freesound API [5] to collect sound material for this research. Our goal is a well-generalized cross-synthesis technique and as such we require a heterogeneous set of sounds for black-box analysis. To achieve this, we made requests to the Freesound API for randomly-generated sound IDs. We used sounds that are lossless, contained one or two channels, had a sample rate of $44.1kHz$, a duration between 0.05 and 5.0 seconds, and were uploaded by a user that was not already represented in the dataset.

We gathered a collection of 1024 sounds satisfying these criteria and henceforth refer to it as the *randomized Freesound dataset* (RFS). We average stereo sounds in RFS to mono and scale all original and hybrid sounds to a peak amplitude of 1 before convolution and analysis.

3.2 Data Preparation

From the 1024 sounds in RFS we generated 512 random pairs of sounds without replacement. We subjected each of these pairs to cross-synthesis via convolution and extended convolution with four different parameter configurations resulting in five sets of 512 hybrid sounds. Using the Essentia software package [6], we perform feature extraction on both RFS and the hybrid sets to analyze what changes convolution yields to acoustic features on average.

We identified the following acoustic features as useful for general analysis: *loudness*, *spectral centroid*, and *spectral flatness*. We computed each of these features for RFS as well as the five hybrid sets listed below. All spectral features were computed using windows of size 1024 with 50% overlap and Hann windowing. Features were averaged across all windows per sound then across all sounds per set.

1. *RFS*: 1024 RFS sounds
2. *OC*: 512 RFS pairs subjected to Ordinary Convolution ($p = 1/2, q = 1, r = 1/2, s = 1$)
3. *GMMC*: 512 RFS pairs subjected to Geometric Mean Magnitude Convolution ($p = 1/2, q = 1/2, r = 1/2, s = 1$)
4. *HPC*: 512 RFS pairs subjected to Half Phase Convolution ($p = 1/2, q = 1, r = 1/2, s = 1/2$)
5. *SMC*: 512 RFS pairs subjected to Skewed Magnitude Convolution ($p = 1, q = 1, r = 1/2, s = 1$)
6. *SPC*: 512 RFS pairs subject to Skewed Phase Convolution ($p = 1/2, q = 1, r = 1, s = 1$)

3.3 Loudness

Raw gain in peak amplitude created by convolving two sources is difficult to predict. Since we are working in the realm of offline cross-synthesis, we ignore the issue and instead focus on perceptual loudness assuming all sounds have been scaled to the same peak amplitude. We hope to use this measure to establish the average effect that convolutional cross-synthesis has on loudness.

Loudness, defined by Steven's power law as energy raised to the power of 0.67 [7], is a psychoacoustic measure representing the perceived intensity of a signal. Loudness of two signals with the same peak amplitude can differ significantly. We calculate loudness for each window of each sound and report loudness for all sets in Table 1.

Set	Mean	Std. Dev.	Min.	Max.
RFS	7.6333	7.8888	0.3095	34.152
OC	7.5306	8.4834	0.0067	41.703
GMMC	5.7503	5.3752	0.3390	28.092
HPC	2.2905	1.9530	0.4358	13.476
SMC	9.6432	9.1378	0.7742	45.805
SPC	6.4163	6.6037	0.6629	40.045

Table 1: Windowed loudness values for all sets.

The mean loudness of all hybrid sets is skewed by the exaggerated tail created by convolution. It is more telling to examine the max loudness. OC produces higher average max loudness than RFS, while GMMC and HPC produce lower max loudness. Both GMMC and HPC have an averaging effect on the amplitude envelope of the result which causes this reduction (as is indicated by their lower standard deviation). SMC and SPC both produce an increase in max loudness compared to RFS that is similar in magnitude to the increase produced by OC.

3.4 Spectral Centroid

The spectral centroid is the barycenter of the magnitude spectrum using normalized amplitude [8]. Listed in Table 2 in Hz, the spectral centroid represents a good approximation of the "brightness" of a sound. The higher the value, the brighter the perceived sound. We use the spectral centroid to quantify our informal observation of high frequency attenuation produced by convolutional cross-synthesis.

Set	Mean	Std. Dev.	Min.	Max.
RFS	3333.3	1467.7	1212.0	7634.0
OC	1206.4	677.18	341.14	4480.5
GMMC	3590.2	745.49	2049.8	6083.7
HPC	1206.6	433.52	803.56	5455.0
SMC	1048.8	351.16	623.38	3842.9
SPC	1156.2	343.68	677.39	3768.9

Table 2: Spectral centroid values for all sets.

The average spectral centroid of the outcome of OC is approximately 64% lower than that of RFS. This confirms our observation that the output of convolution is often perceptually darker than the inputs. GMMC brings the average spectral centroid to a similar level of the original sounds. Both SMC and SPC have a similar effect on the perceived brightness compared to OC, indicating that our source influence parameters (p, r) are relatively independent from our parameter controlling brightness (q).

3.5 Spectral Flatness

Spectral flatness is a measure of the noisiness of a signal and is defined as the ratio of the arithmetic mean to the geometric mean of spectral amplitudes [8]. The measure approaches 1 for noisy signals and 0 for tonal signals. Spectral flatness values for all sets appear in Table 3. We use

this measure to demonstrate our informal observation that ordinary convolution overemphasizes constructive spectral interference yielding results that are less flat than the inputs.

Set	Mean	Std. Dev.	Min.	Max.
RFS	0.2158	0.1132	0.0628	0.5778
OC	0.0207	0.0299	0.0015	0.2761
GMMC	0.2649	0.0670	0.1289	0.5192
HPC	0.0191	0.0550	0.0040	0.6240
SMC	0.0073	0.0320	0.0005	0.4047
SPC	0.0163	0.0310	0.0026	0.3972

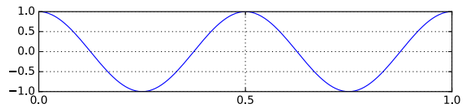
Table 3: Spectral flatness values for all sets.

The difference between the spectral flatness for OC and GMMC is pronounced; the average spectra for the product of GMMC is roughly 13 times flatter than that of OC. SMC further emphasizes constructive interference as it is a self-multiplication and produces even less flat results than OC. HPC and SPC have a less notable effect on spectral flatness as the measure does not consider phase.

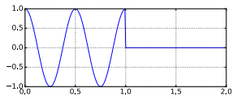
4. CIRCULAR ARTIFACTS

Nonlinear manipulation of magnitude and phase components in the frequency domain via extended convolution has particular side effects in the time domain.

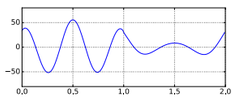
In Figure 2 we show a 128-sample sinusoid undergoing convolution with a unit impulse using $q = 1$ (ordinary) and $q = 2$ (squared magnitude). We see that ordinary convolution preserves the precise arrangement of frequency components that allow zero-padding to be reconstructed with the inverse Fourier transform, while squaring the magnitude spectra does not. Instead, the onset shifts circularly and has an unpredictable amplitude envelope preventing us from discarding zero-padded samples.



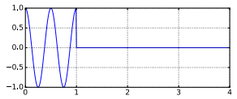
(a) Input $N = 128$



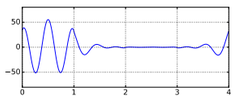
(b) $NFFT = 256$, $q = 1$



(c) $NFFT = 256$, $q = 2$



(d) $NFFT = 512$, $q = 1$



(e) $NFFT = 512$, $q = 2$

Figure 2: Example of circular phenomena when convolving a sinusoid with a unit impulse using $q = [1, 2]$.

These types of artifacts are always cyclical and often produce musically interesting results. With extreme parameter configurations for extended convolution, the DFT size (which must be greater than or equal to the sum of the input lengths less one) can be reinterpreted as a parameter that affects the length of the result. This can produce a desirable effect of sustained, ambient timbres especially when using high s values. Unfortunately, these artifacts also prevent a real-time implementation of extended convolution using partition methods. This is an area for future investigation but is not critical for the purpose of cross-synthesis.

5. CONCLUSIONS

We have demonstrated an extended form of convolution for offline cross-synthesis that allows for parametrized control over the result. Through our extensions to convolution, a musician interested in cross-synthesis now has control over brightness as well as independent control over source emphasis in both magnitude and phase. We have also shown that our extensions influence acoustic features of hybrid results in a meaningful way. Cross-platform software implementing the techniques described in this paper can be obtained at `http://chrisdonahue.github.io/ject`.

Acknowledgments

This research is supported in part by the University of California San Diego General Campus Research Grant Committee and the University of California San Diego Department of Music. Thanks to the Freesound team for their helpful project, and to the anonymous reviewers for their constructive feedback during the review process.

6. REFERENCES

[1] M. Dolson, *Recent advances in musique concrete at CARL*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 1985.

[2] M.-H. Serra, *Musical signal processing*. Routledge, 1997, ch. Introducing the phase vocoder.

[3] C. Roads, *The computer music tutorial*. MIT press, 1996.

[4] T. Erbe, *PVOC KIT: New Applications of the Phase Vocoder*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2011.

[5] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proceedings of the 21st acm international conference on multimedia*. ACM, 2013, pp. 411–412.

[6] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra, "Essentia: An Audio Analysis Library for Music Information Retrieval." in *ISMIR*, 2013, pp. 493–498.

[7] S. S. Stevens, *Psychophysics*. Transaction Publishers, 1975.

[8] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," 2004.

Algorithmic Composition in Abjad: Workshop Proposal for ICMC 2016

Trevor Bača, Harvard University;
Jeffrey Treviño, Colorado College;
Josiah Oberholtzer, Unaffiliated

Abjad is an open-source software system designed to help composers build scores in an iterative and incremental way. Abjad is implemented in the Python programming language as an object-oriented collection of packages, classes and functions. Composers visualize their work as publication-quality notation at all stages of the compositional process using Abjad's interface to the LilyPond music notation package. The first versions of Abjad were implemented in 1997 and the project website is now visited thousands of times each month.

In the context of the primary themes of ICMC 2016 — "Is the sky the limit?" — the principle architects of Abjad propose to lead a hands-on workshop to introduce algorithmic composition in Abjad. Topics to be covered during the workshop include: instantiating and engraving notes, rests, chords; using the primary features of the Python programming language to model complex and nested rhythms; leveraging Abjad's powerful iteration and mutation libraries to make large-scale changes to a score; and introducing the ways composers can take advantage of open-source best practices developed in the Python community.

Abjad is a mature, fully-featured system for algorithmic composition and formalized score control. Because of this we are able to work flexibly with ICMC conference organizers as to the duration of this workshop. We have given both 45-minute and three-hour versions of similar workshops before. So we leave the duration of this proposal open and we invite conference organizers to suggest a duration for the workshop that best fits the conference schedule.

Topics: **algorithmic composition** and **composition systems and techniques**.

Workshop-proposal for ICMC2016 “The Art of Modelling Instability in Improvisation”

IOM – AIM Research: “The Art of Modelling Instability in Improvisation”.

Proposal by the LOOS Foundation / Studio LOOS.

IOM - Interactive Interdisciplinary Improvisational Orchestral Machine
AIM - Artificial Improvisation Machine

Abstract.
IOM – AIM Research, “The Art of Modelling Instability in Improvisation” presents a workshop (concert & lecture) that invites audience to participate in a sound an visuals emerging environment while researching stability vs instability, communication, expectation, confirmation, surprise, unpredictability, risk taking, de-stabilization, problem finding-creating-solving, creativity.

As a state of the art artistic research, IOM-AIM Research investigates, and offers possibilities to model and transform, primary the unstable aspects of musical improvisation by humans.

IOM - AIM Interactive Interdisciplinary Improvisational Orchestral Machine – Artificial Improvisation Machine, is also an environment as a tool in which artificial improvisers (computers, speakers, projectors, sensors - microphones, ultra-sone distance sensors, cameras) audience (for instance by means of an interactive app) and human improvisers as independent actants interactively cooperate and perform.

The environment is developing its own DNA, has been taught, learns from and communicates with itself, past and current experiences and by doing so, establishes a personal identity, thinking style and way of creating.

In the past, IOM-AIM has in settings with multiple speakers (from 4 to 24), and projectors (1 – 4), collaborated successfully with around 87 human artists: a variety of software programmers, individual improvising and interpreting musicians, graphic designers, visual artists, musical acoustical ensembles and orchestras.

Artistic themes have been Instability, Composing Jazz, Musician’s Profile, The Art of Memory, Exploring the Space, Transforming the Space, interactive Cathedral of Thorns, Mimeses, Concatenation, Some Rules in the Zoo and Communication and Flow.

Keywords: stability, emergence of sound and visuals, communication, expectation, confirmation, transformation, surprise, unpredictability, risk taking, de-stabilization, instability, problem finding-creating-solving, thinking styles, creativity.

CREATE Studio Report 2016

Curtis Roads
CREATE
University of California
Santa Barbara CA USA
clang@create.ucsb.edu

Andres Cabrera
CREATE
University of California
Santa Barbara CA USA
mantaraya36@gmail.com

JoAnn Kuchera-Morin
CREATE
University of California
Santa Barbara CA USA
jkm@create.ucsb.edu

Clarence Barlow
CREATE
University of California
Santa Barbara CA USA
barlow@music.ucsb.edu

ABSTRACT

Embracing the shores of the Pacific Ocean in Santa Barbara, the Center for Research in Electronic Art Technology (CREATE) serves the Music Department and the Media Arts and Technology (MAT) program at the University of California, Santa Barbara (UCSB). The Center provides a dynamic environment for students, researchers, and media artists to pursue research and realize a wide array of works. The UCSB AlloSphere is a unique immersive R&D instrument with a 54.1 Meyer Sound system. Courses are offered at the undergraduate and graduate levels in collaboration with several departments.

1. CREATE HISTORY AND ACTIVITIES

Under the leadership of Professor Emma Lou Diemer, UCSB set up its first electronic music studio in 1972 based around a Moog IIC modular synthesizer [1]. In 1986, Prof. JoAnn Kuchera-Morin founded CREATE with the major acquisition of a Digital Equipment Corporation VAX-11/750 computer and associated peripherals, including audio converters. By the mid-1990s, the VAX computer was replaced by several workstations [2]. The rest of this report concentrates on recent developments (2015-2016).

Administratively, CREATE is situated within two academic units: the Music Department and the MAT graduate program. Prof. Kuchera-Morin is Director, Curtis Roads is Associate Director, and Andres Cabrera is Research Director. Dr. Cabrera replaced Matthew Wright, who has moved on to a similar position at CCRMA, Stanford. Corwin Chair Prof. Clarence Barlow is an affiliate faculty member. Counting MAT students and music composition students together, CREATE serves about 55 PhD and Masters students at any given time.

CREATE functions as a research and development facility available to students, researchers, and professional artists for scientific studies and realization of media art works, including live ensembles.

The Center maintains close ties to the UCSB AlloSphere [3, 4]. The AlloSphere is a three-story-high immersive multiuser interactive visual/aural composition and performance platform. In practice, it serves as a research instrument for composing and rendering large scale *n*-dimensional data sets, running simulations, and solving mathematical equations for data exploration and discovery. The AlloSphere is designed to enable teams of interdisciplinary researchers to work together in conducting large-scale real-time data mining.

Figure 1. Rendering of the UCSB AlloSphere showing one ring of Meyer Sound loudspeakers at the top and four of the 27 video projectors at the bottom. (Two other loudspeaker rings are not shown.) The bridge in the middle can accommodate up to 40 people.

2. EDUCATION

CREATE faculty offer courses in both Music and MAT at the undergraduate and graduate levels. The curriculum includes a unique multi-course sequence in digital audio and media programming. Many alumni have gone on to top industry positions after this training, including David Thall, who directs game audio development at Apple. Others have taken positions at Adobe, Dolby, GraceNote, and Epic Games.

One course teaches students how to make 3D visual and audio content in the AlloSphere. We also offer a one-year introductory course in audio recording and sequencing, digital synthesis, mixing, and signal processing. Advanced courses on special topics such as sound in space and modular synthesis are also offered.

3. RESEARCH PROJECTS

The Gamma Audio DSP Library was developed by PhD student Lance Putnam [5] to provide a C++ library using APIs to the AlloSystem libraries in order to provide a unified system for the development of audiovisual software for the AlloSphere.

PhD student Ryan McGee developed the Sound Element Spatializer, a GUI interface to control spatialization of sound through OSC, and then ported his work to AlloSystem to provide a set of spatializers with a unified interface that include vector-based panning, DBAP, and Ambisonics for AlloSphere audio content [6].

Zirkonium Chords (ZC) is a research project of Curtis Roads and MAT alumnus Chandrasekhar Ramakrishnan. It is based on the Zirkonium spatialization software developed for the ZKM, Karlsruhe [7, 8]. ZC became operational on 25 July 2013. The first composition to be spatialized by ZC in the AlloSphere was *Sculptor* (2001) by Curtis Roads. We developed a new version of ZC in November 2015 for a concert in which Curtis Roads performed a live upmix of his composition *Then* using the 47 loudspeakers of the ZKM Klangdom.

4. CREATE STUDIOS AND LABS

In addition to the AlloSphere managed by Prof. Kuchera-Morin and Dr. Cabrera, CREATE maintains four studio laboratories: Theremin, Xenakis, Varèse, and the Pluriphonic laboratory. The first three are in the Music Building, the last one is housed in Elings Hall. Studio Varèse is our core studio for artistic research and development. It features an octophonic sound system of Dynaudio loudspeakers. Varèse also houses the CREATE Modular Synthesizer, a large Eurorack-based system with some 40 modules. The CREATE Teaching Synthesizer is a smaller 17-unit Eurorack modular synthesizer designed specifically for the modular synthesis course taught by Prof. Roads. The Modular and the Teaching systems can be used independently or combined for exceptionally complex patches. We emphasize analog computing concepts as well as sound synthesis and processing.

Figure 2. CREATE Modular synthesizer includes 40 EuroRack modules from various companies.

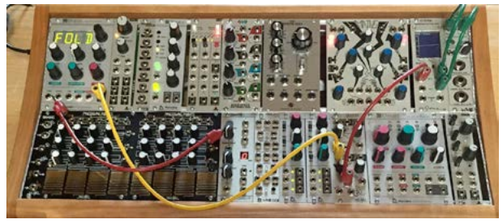


Figure 3. CREATE Teaching synthesizer.

5. COMPOSITION

Recent compositions include *Then* by Curtis Roads, which premiered at the 2014 International Computer Music Conference in Athens (September 2014). It was extensively revised in 2015 and will appear on a vinyl LP by KARL Records, Berlin. Roads also completed *Still Life* based on sounds composed by the late Stephan Kaske and *Modulude* (2016).

In 2015, Clarence Barlow realized three algorithmic works: *)ertur(* for video, flute, clarinet, violin, violoncello and piano, *...until... #10* for double bass and drone, and *Amnon, who led it* for tenor saxophone, violoncello, Hammond organ, and piano four hands.

6. PUBLICATIONS

Oxford University Press published Curtis Roads's new book *Composing Electronic Music: A New Aesthetic* in 2015 [9]. The accompanying web site features 155 sound examples:

http://global.oup.com/us/companion.websites/9780195373240.

7. CONCERTS

Most CREATE concerts take place in Lotte Lehmann Concert Hall (LLCH), a 460-seat theater. Within LLCH, the Creatophone is a permanently-installed 8.1 Meyer Sound system for the spatial projection of music.

A major CREATE concert was held February of 2016 in the UCSB AlloSphere using its 54.1 Meyer Sound system, which provides 14 kW of power in 360 degrees. The concert featured guest composer John Chowning performing his landmark *Turenas* with 3D visuals depicting the sound path in space by Prof. Ge Wang of CCRMA. Also featured was the CREATE Ensemble and several student works

8. CREATE ENSEMBLE

The CREATE Ensemble is a laptop orchestra founded by Matthew Wright. Although Dr. Wright has recently moved to Stanford, he has continued to interact with the group via network. Each piece is a research experiment in live interaction [12, 13, 14, 15, 16]. A highlight was a 2014 network concert with musicians at Stanford University, Virginia Tech and the Universidad de Guanajuato. The CREATE Ensemble's *Feedback II* (2014) and *Feedback IV* (2015) explore a performance paradigm in which each instrument has an audio input that is incorporated into its audio output. A digital patching matrix (visualized for the audience) creates connection topologies among the ensemble by mixing the instruments' outputs to form each instrument's input.

9. INSONIC CONFERENCE 2015 AND ACADEMIC EXCHANGE

CREATE and MAT were co-organizers of the InSonic conference on spatial audio held in Karlsruhe, Germany in November 2015 in collaboration with the University of Media, Art, and Design (HfG), the Center for Art and Media Technology (ZKM), and Ircam. The HfG and CREATE also organized an academic exchange program, sponsored by the Baden-Württemberg Foundation. Two UCSB PhD students, Fernando Rincón Estrada and Muhammed Hafiz Wan Rosli, carried out spatial audio research projects at the ZKM facility.

10. GUESTS AND VISITING ARTISTS

Guest lecturers and visiting artists in recent years have included Richard Devine, Bruce Pennycook, Rozalie Hirs, Henning Berg, Ragnar Grippe, Xopher Davidson, David Rosenboom, Earl Howard, David Wessel, Anke Eckhardt, Markus Schmickler, Tony Orlando (MakeNoise), Peter Castine, Nicholas Isherwood, James Dashow, Thom Blum, Tim Feeney, Vic Rawlings, Kaffe Matthews, Kasper Toeplitz, Hubert S. Howe, Jr., Maggi Payne, Ron Sword, Robert Morris, Yann Orlarey, Philippe Manoury, John Chowning, Max Mathews, and Jean-Claude Risset. CREATE facilities can be made available for guest research and artistic residencies that are self-funded.

Acknowledgments

Our thanks to Matthew Wright for his many contributions as CREATE's former Research Director. Thanks to Michael Hetrick for his advice in configuring the CREATE synthesizers and to Sekhar Ramakrishnan for his assistance with Zirkonium Chords. We thank Paul Modler for organizing the academic exchange between Santa Barbara and Karlsruhe.

11. REFERENCES

[1] Diemer, E. L. 1975. "Electronic Music at UCSB." *Numus-West:* 56-58.

[2] Kuchera-Morin, J., C. Roads, A. de Campo, A. Deane, S. Pope. 2000. "CREATE Studio Report 2000." *Proceedings of the International Computer Music Conference 2000.* San Francisco: International Computer Music Association. pp. 436-438.

[3] Amatriain, X., T. Höllerer, J. Kuchera-Morin, and S. Pope. 2009. "The Allosphere: immersive multimedia for sscientific description and artistic exploration." *IEEE Multimedia* 16(2): 64-75.

[4] Cabrera, A., J. Kuchera-Morin, C. Roads. 2016. "The evolution of spatial audio in the AlloSphere." *Computer Music Journal.* In Press.

[5] Putnam, L. 2014. "Gamma: A C++ Sound Synthesis Library Further Abstracting the Unit Generator." *Proceedings of the International Computer Music Conference-Sound and Music Computer Conference 2013.* Athens: International Computer Music Association. pp. 1382-1388.

[6] McGee, R. Wright, M. 2011. "Sound Element Spatializer." *Proceedings of the International Computer Music Conference 2011.* Huddersfield: International Computer Music Association.

[7] Ramakrishnan, C. 2007. *Zirkonium.* Karlsruhe: ZKM Institut für Musik und Akustik.

[8] Ramakrishnan, C. J. Gossmann, and L. Brümmer. 2006. "The ZKM Klangdom." Paris: *Proceedings of NIME 06.*

[9] Roads, C. 2015. *Composing Electronic Music: A New Aesthetic.* New York: Oxford.

[10] Roberts, Charlie, Jesse Allison, Ben Taylor, Daniel Holmes, Matthew Wright, JoAnn Kuchera-Morin. 2015. "Educational Design of Live Coding Environments for the Browser." In *Journal of Music, Technology and Education* (special edition). In press.

[11] Roberts, Charles, Graham Wakefield, Matthew Wright, and JoAnn Kuchera-Morin. 2015. "Designing Musical Instruments for the Browser." *Computer Music Journal* 39:1, pp.27-40. Spring 2015.

[12] Roberts, Charles, Matthew Wright, and JoAnn Kuchera-Morin. 2015. "Beyond Editing: Extended Interaction with Textual Code Fragments." In *Proceedings of the International Conference on New Interfaces for Musical Expression.* Baton Rouge, LA.

[13] Roberts, Charles, Matthew Wright, JoAnn Kuchera-Morin, and Tobias Höllerer. 2014. "Gibber: Abstractions for Creative Multimedia Programming." In *Proceedings of the ACM International Conference on Multimedia.* New York, NY, Pages 67-76.

[14] Roberts, Charles, Matthew Wright, JoAnn Kuchera-Morin, and Tobias Höllerer. 2014. "Rapid Creation and Publication of Digital Musical Instruments." In *Proceedings of the International Conference on New Interfaces for Musical Expression.* London, UK.

[15] Wan Rosli, Muhammad Hafiz, Karl Yerkes, Matthew Wright, Timothy Wood, Hannah Wolfe, Charlie Roberts, Anis Haron, and Fernando Rincón Estrada. 2015. "Ensemble Feedback Instruments." In *Proceedings of the International Conference on New Interfaces for Musical Expression.* Baton Rouge, LA.

[16] Yerkes, Karl, and Matthew Wright. 2014. "Twkyr: a Multitouch Waveform Looper." In *Proceedings of the International Conference on New Interfaces for Musical Expression.* London, UK.

Computer Music Studio and Sonic Lab at Anton Bruckner University Studio Report

Andreas Weixler
Anton Bruckner University
Computer Music Studio
Linz, Austria/EU
a.weixler@brucknerUni.at

Se-Lien Chuang
Atelier Avant Austria
Austria/EU
chuang@mur.at

ABSTRACT

The CMS (Computer Music Studio) [1] at Anton Bruckner University in Linz, Austria is hosted now in a new building with two new studios including conceptional side rooms and a multichannel intermedia computer music concert hall - the Sonic Lab [2].
The Sonic Lab is one of the three concert halls at the new campus building of the Anton Bruckner University. It is designed as a computer music concert hall dedicated to multichannel computer music and electroacoustic music, as well as experimental music in cooperation with JIM (the Institute of Jazz and Improvised Music), among others. The development of the CMS is based on an initiative of Ao.Univ.Prof. Andreas Weixler during the years 2005 - 2015 who drafted a plan for a suite of rooms for the Computer Music Studio: Sonic Lab - multichannel computer music concert hall (20.4), Production Studio (20.2), Lecture Studio (8.1), Research Zone (4.1), Project Room (4 ch), Archive, Workshop, Machine Room and two faculty offices.

1. INTRODUCTION

The colloquium of the Computer Music Studio has been offering lectures and courses in the field of music and media technology, media composition and computer music since its formation 1995 as SAMT (Studio for Advanced Music and Media Technology). The range of subjects it offers is closely integrated with those of the former Institute DKM (Composition, Conducting and the Theory of Music), which since October 2015 has been divided into two Institutes IKD (Institute of Composition and Conducting) and ITG (Institute of Theory and History of Music), as well as JIM. The CMS can be seen as an interface and competence centre spanning several institutes of the Bruckner University, active in the region through numerous co-operations and internationally networked with exchanges and a lively conference scene. The Computer Music Studio organizes numerous concert and lecture series, regionally, nationally and internationally [3].

1.1. History

The Computer Music Studio was founded in 1995 as the SAMT - Studio for Advanced Music And Media Technology by DI Adelhard Roidinger and the Rector of the Bruckner Conservatory, Hans Maria Kneihs in the buildings of the Software Park Hagenberg.
The staff consisted of three teachers and a technician: Adelhard Roidinger, Karlheinz Essl, Andreas Weixler and Gerald Wolf. The staff was then gradually reduced to one person and the studio lost all its budgets. Since 2008 the University studio, as it became, has been under the direction of Ao.Univ.Prof. Andreas Weixler, who was at that time the only member of staff in an environment concerned with media archaeology (sic!); this firstly prevented the studio from being closed and, secondly, helped restore funding and activities. At the same time an institute directive changed the name of the studio to CMS (Computer Music Studio). In October 2015, Univ. Prof. Volkmar Klien was appointed to a new professorship with emphasis on media composition and computer music. Prior to moving into the new building in September 2015 the CMS consisted of 3 rooms, a lecture studio, a production studio and an office/archive. With the new premises the Bruckner University caught up on international standards, after 10 years of efforts by the authors.

2. THE FACILITIES

The Computer Music Studio has been proudly conceived, constructed and expanded under the direction of Andreas Weixler throughout 10 years (2005-2015) of negotiations with the university direction. Now it comprises a prestigious computer music concert hall named Sonic Lab with an adjacent production studio, a lecture studio (Lehrstudio), a project-oriented space with pesonalized working desks (Projektraum), an experimental research zone (Computermusik-Forschungsraum), a workstation, an archive room/depot and (last but not least) offices for colleagues and the directors.

2.1. Sonic Lab

The Sonic Lab is an intermedia computer music concert hall with periphonic speaker system, created by Andreas Weixler for the Bruckner University to enable international exchanges for teaching and production with other developed computer music studios. 20 full range audio channels plus 4 subsonic channels surround the audience, enabling sounds to move in space in both the horizontal and vertical planes. A double video and data projection capability allows the performance of audiovisual works and also the accommodation of conferences, etc.

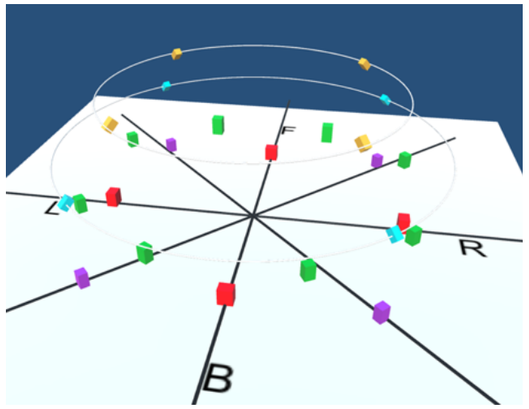


Figure 1. 3D model of the speaker array [4]

The speciality of the CMS is interactive audiovisual performances, in which the sound of acoustic musical instruments produce images and spatial sounds in interplay. However, the Sonic Lab is also a perfect venue for concerts of jazz and improvised music (which often features strong percussive and amplified sounds) and for contemporary music (which frequently requires dry and clear acoustics) thanks to its special acoustic treatment.. The Sonic Lab is thus a place for the music of the future.

2.1.1. Opening Sonic Lab [5]
The opening ceremony took place on 17th of November 2015 with concerts and workshops featuring honorary guests John Chowing (Emeritus Professor at Stanford University), Jonty Harrison (BEAST, Emeritus Professor at Birmingham University), Karlheinz Essl (University of Music and Performing Arts, Vienna), Gerfried Stocker (ars electronic centre Linz) organized by Andreas Weixler and Se-Lien Chuang. Keynotes were given by Jonty Harrison (*Tuning In to the Future*) and John Chowning (*Loudspeakers as Spatial Probes*).

Figure 2. preparation at the Sonic Lab

In the opening concert compositions by John Chowning (*Voices* - for Maureen Chowning - v.3 for soprano and electronics), Jonty Harrison (*BEASTiary*), Karlheinz Essl (*Autumn's Leaving* for pipa and live electronics), Se-Lien Chuang (*Nowhereland* for extended piano, bass clarinet, multichannel electro-acoustics and live electronics) and Andreas Weixler (*Wetterleuchten - Virtuoso Chances - return home* - video with algorithmic multichannel electroacoustic music), Hassan Zanjirani Farahani (*Das Unlogische notwendig* for soprano, live electronics and light design) were performed by Maureen Chowning (soprano), Ming Wang (pipa), Elvire De Paiva e Pona (soprano), Julia Lenzbauer (clarinet) and Mariia Pankiv (piano). The finale was *Momentum Opening Sonic Lab*, a group improvisation & interactive audiovisual transformation with all-star performers. The following two days of workshops were open to the public: Jonty Harrison (*Final*

Figure 3. Momentum Opening Sonic Lab [6]

Frontier or Open Border) and John Chowning (*Sound Synthesis and Perception: Composing from the Inside Out*). The demand for the event by over 100 people was much higher then the capacity of the Sonic Lab, where 55 people can listen in optimal conditions; nevertheless, 80 people squeezed in and the opening events were transmit-

ted to another concert hall in the building, with an eight channel surround sound and a live video transmission. We were very honoured by the presence of Laurie Anderson and Dennis Russell Davies in the audience.

2.2. Multichannel Concept

The teaching studio (Lehrstudio), the production studio and the Sonic Lab itself are all equipped with compatible devices based on a Digidesign C24 DAW controller and 32 channel Protools system as digital mixing console, playback and recording device, together with basic software including ProTools, Ableton Live, Cycling '74's MaxMSP and Final Cut Pro.

2.2.1. Teaching Studio (Lehrstudio)

This has a circle of eight Genelec 1032 speaker and a Genelec 7070 sub. The studio has an on site machine room to keep noise out of the studio.

Figure 4. teaching studio (Lehrstudio) [7]

2.2.2. Production Studio

This has a circle of eight Genelec 1032 speaker and 2 Genelec 7070 sub as well as 12 more speakers, creating a sound dome in 3 quadraphonic layers:
4x Genelec 8040A in 3 m, hi level speaker
4x Genelec 8040A in 5,5 m, ceiling speaker
4x KS Audio CPD 12M, on ground level

2.2.3. The Sonic Lab - Computer Music Concert Hall

This has a ring of 8 Genelecs (two 1037B at the front and six 1032A), all at a height of 1.5 m,
plus 4 Genelec 7070 subs (one on each wall) together with 12 more speakers, to create a sound dome in 3 quadraphonic layers corresponding to the production studio:
4x Genelec 8040A at a height of 3 m (high level speakers); 4x Genelec 8040A at a height of 5,5 m (ceiling speakers); 4x KS Audio CPD 12M, at ground level.

There also is a PA System comprising:
2x Kling & Freitag CA1215-9SP
2x Kling & Freitag SW 115-E SP
4x KS Audio CPD 12M (monitors)
equipped with a 24 channel analog Soundcraft GB 4 and a digital Yamaha 01V mixing console.

Throughout the building there are rings of MADI lines which can connect to the sound recording studioTON, located in the basement of the university, and to its other concert halls, as well as to the rooms of JIM.

2.3. Replica Room - The Production Studio

As the usage of the Sonic Lab is also for the JIM and open to everyone in the building, as with all the concerts-halls at Bruckner university, the production studio is a complete replica of the Sonic Lab, where you can work for longer period with the same settings as in the Sonic Lab. Both share a machine room to accomplish silence.An acoustically treated window between the two rooms create a great combination of production and performing venue.

2.4. Project Room

This room aims to develop social interaction and the sharing of collective know-how between CMS students. It has six computer workstations with iMacs and basic software including ProTools, Ableton Live, Cycling '74's MaxMSP and Final Cut Pro. Workstations can be dedicated to individual students or, if necessary, a group.

2.5. Experimental Research Zone

This is dedicated to future PhD candidates. In the meantime, it has combined functions: as a place for experimental work (enabling set-ups to be left for longer periods of research), a laptop lecture room and a second social room where students can work freely.

2.6. Workshop (Werkstätte)

This is a room for soldering, repair and construction .
It also contains a media archeological workstation to read out dated formats such as DAT, VHS, ZIP, JAZ, SCSI, etc.

2.7. Depot / Archive

As well as storing the equipment of the CMS computer music studio, this has computer facilities for the digital archive and also provides a physical archive for scores, flyers etc.

3. THE STUDIES

Since its start in 1995, music and media technology was the main aim of former SAMT (Studio for Advanced Music and Media Technology). In 2008 when the Studio was renamed CMS (Computer Music Studio), graduate students - mainly in Jazz and improvised music among others - could choose an emphasis in music and media technology for their degree as well as a pedagogical Masters in music and media technology.

In 2014, a new Bachelors program was established in media composition and computer music.
The four-year study includes:

- composition in computer music, intermedia works
- performance computer music, machine musicianship
- production and programming

4. CMS SERIES

Since its new incarnation in 2008 under the direction of Andreas Weixler, the CMS has initiated several series of concerts and exchanges:

4.1. SonicIntermedia

This series of intermedia concerts and lectures for international exchange in Deep Space [8] at the Ars Electronica Center started in 2009. Sonic Intermedia is a media concert series initiated by the composers and media artists Andreas Weixler and Se-Lien Chuang with the artistic director of the AEC, Gerfried Stocker, to give contemporary intermedia computer music a presentation format in Linz. With Sonic Intermedia a new concert format for intermedia art of sound is presented as a co-operation between the Anton Bruckner University and the Ars Electronica Center. The intention of SonicIntermedia is to create exciting concerts of experimental computer music and intermedia composition with a renowned team of composers, researchers, media artists and musicians.
Guests of SonicIntermedia have included:
2009 SARC: Pedro Rebelo (Portugal/UK), Franziska Schroeder (Germany/UK), Imogene Newland (UK), David Bird (UK), Brian Cullen (UK), Orestis Karamanlis (Greece/UK); 2010 UEA: Simon Waters, Ed Perkins, Bill Vine, Anton Lukoszevieze (UK) and a piece by Nick Melia & Ed Kelly; 2012 BEAST: Jonty Harrison, Scott Wilson, Chris Tarren und Tim Moyers; 2013 NOVARS:
David Berezan, Constantin Popp, Mark Pilkington and pieces by Manuella Blackburn and Claudia Larcher.

4.2. Sound & Vision

This is an experimental series for students and artistic research projects across institutions and cooperating universities. Events to date: Sound and Vision I - AVI: with the ensemble for new music and audiovisual interactivity, Linz; Sound and Vision II - musical inspiration and digital concepts: interaction and improvisation with students of Andreas Weixler; Sound and Vision III - Double Concerto: Electronics and three pianos. Concert of music and media technology with interaction, improvisation and composition with students of Martin Stepanik and Andreas Weixler; Sound and Vision IV - KUNST:MUSIK in cooperation with the University of Arts and Industrial Design, Linz; Sound & Vision V - concert of music and media technology and alternative concepts for ensembles; Sound & Vision VI - intermedia concerts of CMS students; Sound & Vision VII - InterAct: computer music and intermedia concert, pieces with sensor technology, Kinect and others; Sound & Vision VIII - react: computer music with students and guest of the CMS; Sound & Vision IX - Interface: intermedia concert in cooperation with InterfaceCulture students of University of Arts and Industrial Design, Linz; Sound & Vision X - Periphonic-Sonic: first Sound and Vision concert in the Sonic Lab using the periphonic multichannel sound system.

4.3 CMS Invited Lectures Series

This is the international lecture series of the Computer Music Studio at the Bruckner University for external guests. So far we have hosted CMS invited lectures up to #27: 2009: Pedro Rebelo, SARC - Sonic Arts Research Center, Belfast, Northern Ireland; 2010: JYD - Julian Weidenthaler, Linz, Austria; Dr. Simon Waters, UEA Studios, University of East Anglia, Norwich, UK; 2011: Martin Kaltenbrunner, InterfaceCulture, Kunstuniversität Linz; Thomas Grill - Austrian Research Institute for Artificial Intelligence (OFAI) and Institute of Composition and Electroacoustics at the University of Performing Arts and Music, Vienna; André Bartezki, co-director of the electronic music studio at the TU-Berlin; 2012: Jonty Harrison and Scott Wilson, BEAST - Birmingham ElectroAcoustic Sound Theatre, The Electroacoustic Music Studios at the University of Birmingham, UK; 2013: João Pais, composer, Portugal; David Berezan, Electroacoustic Music Studios, Novars, University of Manchester and MANTIS, Manchester Theatre in Sound, University of Manchester, UK; Constantin Popp, PhD research student at Novars, University of Manchester, UK; Mark Pilkington; composer and performer of electroacoustic music, Novars, University of Manchester, UK; 2014: Gonzalo Díaz Yerro, Conservatorio Superior de Música de Canarias; Andrew Lewis, Bangor University, Wales/UK; Mike Frengel, Northeastern University, Boston, USA; Tony de Ritis, Northeastern University, Boston, USA; Sebastian Frisch, freshmania; 2015: Andreas Zingerle, University of Arts and Industrial Design, Linz, Time-based and Interactive Media; Tristan Murail, composer, Paris; Jonty Harrison; BEAST (Birmingham ElectroAcoustic Sound Theatre), Emeritus Professor of the University of Birmingham;

John Chowning, Emeritus Professor at Stanford University; Seppo Gründler, FH Joanneum, Graz; 2016: Christian Tschinkel, composer, Vienna; Rose Soria & Constantin Popp, University of Manchester and Liverpool Hope University, UK.

4.4 CMS Lecture Concerts (Gesprächskonzerte)

This series of events introduces a guest to present concepts and compositional work:
2011 #1 André Bartezki
2014 #2 Gonzalo Yerro
2014 #3 Andrew Lewis
2014 #4 Mike Frengel "prepared guitar and electronics"
2015 #5 Seppo Gründler "Once I was a guitarist"
2016 #6 Christian Tschinkel "The Kuiper Belt Project"
2016 #7 Rosalía Soria & Constantin Popp

4.5 CMS Research Residency

The CMS Research Residency program allows external artists to work with professors and students of the CMS to explore the arts, new concepts, technologies and interactions. The projects so far have been: 2013: Sina Heiss (A/NYC), Lia Bonfilio (NYC), Daniel Rikker (A), "display - mind leap", Tanz, interactive visuals, Stimme, live processing, skeleton tracking; 2014: The CMS CEUS Projekt, research week for students of computer music, music and media technology and guest. This was a workshop in open access for visitors to explore the possibilities of the famous Bösendorfer CEUS grand piano. It was a collaboration of between CMS and Austrian piano manufacturer Bösendorfer hosted by the piano house Merta. The final concert and installation works were by and with: Hassan Zanjirani Farahani, Michael Enzenhofer, Se-Lien Chuang, Andreas Weixler, Daniel Rikker, Thomas Ecker, Martin Stepanik, Elvire de Paiva e Pona and Barbara Mayer; 2015: CMS - composer in residence Jens Vetter (Berlin/Linz) with his project of DMX controlling, video tracking and interactive sound generation; 2016: Christian Tschinkel, „The Kuiper Project" for the Sonic Lab, 2-Track sound distribution for the Sonic Lab.

4.6 CMS Instructional Series

This is a forum for students, undergraduates and professors of ABPU dedicated to a specific topic related to computer music and production to share the in-house expertise. The Admission is free and open to the public. Series I - 2013: *Software, die sich wie ein Instrument spielt* Live Tutorial by Daniel Rikker giving a introduction in Ableton Live - performance, sequencer and producer software. Series II - 2015: Hassan Zanjirani Farahani *Realtime Processing* showing audio realtime processing and performance in Ableton Live and Max. Series III - 2015: Hassan Zanjirani Farahani, technical realization of the composition - *Das unlogische Notwendig* for Soprano, live-electronics und interactive DMX light design; Series IV - 2015/2016: *CMS Sonic Experimental Demos* Introducing the Sonic Lab and Live Processing with Andreas Weixler, Se-Lien Chuang and CMS students, as well as the demonstration of multi-channel works by Jonty Harrison, John Chowning, Fernando Lopez-Lezcano and Christian Tschinkel among others.

4.7 Cooperations

The CMS is cooperating with and supporting other institutes and institutions. *JIM communicate*, for example, is a monthly concert series with faculty and students of the Institute of Jazz and Improvised Music, but also projects with the actors and the dulcimer class are on the list, among many others. There are currently co-operations with and connections to many institutions. Internally, these include: the former Institute for Composition, Conducting and the Theory of Music (DKM), Institute for Jazz and improvised Music (JIM), Institute for Theatre and Drama (ACT), Institute for Keyboard instruments (TAS), Institute of String Instruments (SAI), Institute for Music Education (EMP). One of our goals is to be a competence centre for computer music in the region in cooperations so far with the AEC - Ars Electronic Center, University of Arts and Design, Linz, InterfaceCulture, JKU - Johannes Kepler University, ElisabethInnen Hospital, SCCH Software Park Hagenberg, Klanglandschaften (Soundscapes), Musik der Jugend (Youth Music), Province of OÖ, DorfTV, Klavierhaus Merta. The CMS is also reaching out to other related institutions in Austria, such as the ELAK (Institute for Composition and Electroacoustics, Vienna), MDW (University for Performing Arts and Music, Vienna), Prima la Musica, Salzburg and the piano manufacturer Bösendorfer. Through the personal contact and art work of the authors and a lively connection to international conferences we created a series of international cooperations with JSEM (Japanese Society for Electro Acoustic Music), TU Studio, Berlin, SARC (Sonic Arts Research Centre) Queens University Belfast, Northern Ireland, University of East Anglia, UK, BEAST (Birmingham ElectroAcoustic Sound Theatre), University of Birmingham, UK, NOVARS, University of Manchester, UK), Hope University Liverpool, UK, Northeastern University, College of Arts, Media and Design, Boston, USA, Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, California.

4.8 Students Works

Beside a large number of pieces in the field of electroacoustic composition, interactive and algorithmic composition, sensor technology and light design by CMS students in the field of contemporary composition, beat music, vocal, instrumental and electronic music within audiovisual realtime processing and also time based music and video, there are outstanding Bachelors and Masters theses recently published in the library of the Bruckner University: *The score at the touch of a button* - (*Die Partitur auf Knopfdruck? Computergestützte algorithmisch notierte Komposition mit MAX und LilyPond*), a master thesis 2015 by Michael Enzenhofer, as well as Daniel Rikker´s Masters thesis *Hands on Max for Live*, a great introduction to learning Max4Live and Ableton Live - and emphasis Kinect driven music - last but not least Christoph Hörmann´s very informative *Mixing with Digital Audio Workstations - A guide for home recording musicians* Bachelors thesis in 2015, among a large number of others. Outstanding pieces worth mentioning are *Highway Lights* for drums, four channel audio and light design by Markus Rappold 2015, installation for the CEUS grand piano by Michael Enzenhofer *Wahrscheinliche Wahrscheinlichkeiten* and the above mentioned composition by Hassan Zanjirani Farahani - *Das unlogische Notwendig for Soprano*, live-electronics und interactive DMX light design.

5. OUTLOOK

With the new facilities of the Computer Music Studio, its outstanding arrangement of multichannel computer music studios and the acoustically specialized Sonic Lab, as well as the new appointments of Volkmar Klien as professor of electroacoustic composition and Carola Bauckholt as professor of composition for music theater, in combination with the long time existing professor for music and media technology, computer music and electroacoustic music and director of the Computer Music Studio (CMS) Andreas Weixler, we are looking forward to a new generation of students and a continuing wide range of cooperations, and we hope to contribute with our artistic work, research and educational offerings to the music of the future.

6. REFERENCES

1. CMS - Computer Music Studio https://www.bruckneruni.at/en/Institutes/Conducting-Composition-Theory-of-Music/Computer-Music-Studio
2. The Sonic Lab http://avant.mur.at/weixler/bpu/CMS/SonicLab/index.html
3. CMS concert and lecture series http://avant.mur.at/weixler/studinfo/studinfo_events.html
4. Unity programming by Michael Enzenhofer 2015 http://www.michael-enzenhofer.at/unity3d/SonicLab24ch_web_4/SonicLab24ch_web_4.html
5. Opening Sonic Lab http://avant.mur.at/weixler/bpu/CMS/SonicLab/OPENING/index.html
6. Group Improvisation OSL, © Shayan Kazemi
7. lecturing studio showing Hassan Zanjirani Farahani (©).
8. http://www.aec.at/center/en/ausstellungen/deepspace/

Design Considerations of HKBU's *Laboratory for Immersive Arts and Technology*: a Studio Report

Prof. Christopher Keyes
Hong Kong Baptist University
ckeyes@hkbu.edu.hk

Dr. Christopher Coleman
Hong Kong Baptist University
coleman@hkbu.edu.hk

Vanissa LAW Wing Lun
Hong Kong Polytechnic Univ.
vanissalaw@gmail.comk

ABSTRACT

This paper gives an overview of a recently opened facility dedicated to 3D sound and multi-screen video. Comprising two rooms and a sound-lock, it houses a control room and a theatre that can be configured as a live room, housing the region's only 24.2 channel sound system and 5 permanent HD video screens. At roughly 200m^3 or 70m^2 it is a relatively small facility but has many uses. The state-of-the-art facility has been designed for uniform frequency response and decay rates and a low noise floor. In its construction we were afforded a wide range of possibilities for spatial configurations and equipment choice. Thus a great deal of time was dedicated to exploring various options. It is hoped that presenting some detail on the overall design, including choices available and ultimately implemented, may be of some use for readers planning and budgeting their own facilities, and the issues faced.

1. INTRODUCTION AND CONTEXT

Since early in its history, the Music Department of Hong Kong Baptist University (HKBU) has emphasized music technology in both facilities and curriculum. Its *Electro-acoustic Music Centre* (EMC) opened in 1980, houses 2 control rooms with a shared live room, designed with professional standard acoustics and equipment (see fig. 1).

Figure 1. HKBU's EMC Studio C used by year-2 undergraduates.

The EMC is augmented by a 25-seat laboratory of computers, converters, and headphones. Both serve all undergraduate students with stereo, 5.1, and 6.0 projects. Unfortunately, as space is a rare commodity in Hong Kong, the EMC has a rather small live room that comfortably accommodates solo and chamber music but not larger ensembles. As a single-story structure, it cannot support 3D sound, with microphones and loudspeakers overhead.

The *Laboratory for Immersive Arts and Technology* (LIATe, after the 'LIATE rule' in calculus) is a more recent facility opened in late 2011. Funded largely through a grant from the Hong Kong government, it is a joint university facility for research and creative projects in immersive 3D audio and multi-screen video. It serves mostly university faculty and research postgraduate students (MPhil and PhD) and occupies a two-story space, allowing ample room for overhead speakers and 3 dimensional microphone arrays, and can accommodate larger ensembles than our EMC (see fig. 2). It comprises two rooms and a sound-lock: an industry-standard 7.1 control room, and a more experimental theatre/live room housing the 24.2 channel sound system (see fig. 3). The space is also used for presentations, demonstrations and small-scale performances, and can seat 30 people.

Figure 2. *Laboratory for Immersive Arts and Technology* Theatre.

Its initial proposal followed shortly after the 2007 Digital Cinema Specifications which states: "The delivered digital audio, contained within the Digital Cinema Package (DCP), shall support a channel count of sixteen full-bandwidth channels." ... later specified as 24 bit and up to 96Khz uncompressed '.wav' format.[1] The document does not specify much beyond that, which leaves many interesting questions unanswered. Where should the speakers for those channels be located, how does one record an orchestral film-score intended for 16-channel diffusion, how it will upmix older formats, etc.? These are some of the purely research-oriented questions the lab was built to answer, as well as realizing creative works with a plethora of creative possibilities. The facility also answers some practical problems in Hong Kong, chiefly the lack of a mastering facility for multi-channel audio and a recording studio for 3D audio.

[1] Section 3.3.2.4. Digital Cinema Initiatives LLC, *Digital Cinema System Specification* Version 1.1 April, 2007.

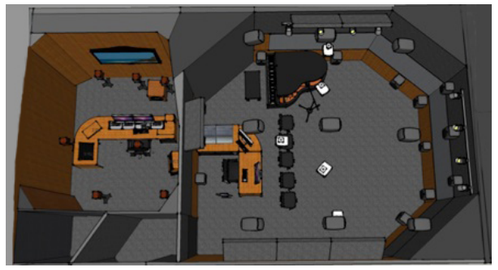


Figure 3. 3D models of HKBU's *Laboratory for Immersive Arts and Technology*; its control room on the left, and theatre on the right.

2. FACILITY DESIGN

2.1 Ambient noise

Naturally one of the first concerns in the planning stage was ambient noise control. Our team had visited another facility with 3D audio and video, however the noise from the projectors rendered the audio almost inaudible. The problem was greatly compounded by the noise from an even louder air conditioning system. To avoid these problems roughly 25% of the renovation costs were for a dedicated off-site humidity and air conditioning control system (HVAC) exclusive to our facility, with baffled airways minimizing noise. This, in combination with a typical studio double-wall construction for sound isolation, left the control room with a NR 15 rating (15dB SPL of ambient noise) and the theatre with a NR 25 rating. As most audio equipment is manufactured for a minimum of noise output, keeping the noise floor low in the control room was not a substantial issue. We did opt for a flat-screen video monitor instead of a projector because of noise concerns, as 3D projectors generally produce more noise than 2D projectors. For the theatre however, which houses 5 projectors, multiple computers, and other video equipment for which noise output is less of a manufacturing concern, ambient noise control took much more careful planning, as detailed below.

2.2 Control room design

The basic design and acoustics of the control room follow current industry models: no parallel surfaces, a balance of absorptive and diffusive materials, careful attention to the dimensions of the studio and first reflections, etc. One distinctive feature however, lies in the handling of bass frequencies, which are often pronounced and difficult to control in smaller rooms. Looking at the model (see fig. 3) one may note that the shape of the control room space is not symmetric. In order to insure the space was symmetric around the listening position, a wall was erected behind the listening position for this purpose. Crucially however, a gap was left (not shown) between the top of the wall and the ceiling to trap bass frequencies in the chamber behind the door. On our first acoustic measurements we were pleased to find that bass frequency response was well controlled and very little additional tweaking was necessary (see fig. 4).

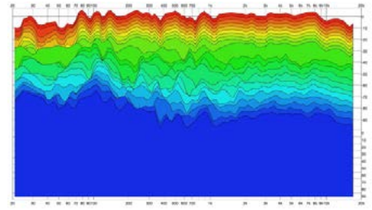


Figure 4. 1/3rd octave cumulous spectral decay plot (waterfall) of the control room's acoustics showing a mostly uniform response and decay rate of frequencies.

2.3 Control room equipment design

The control room design also follows current trends in that most mixing is done in software, augmented with outboard gear for specific colors/effects difficult to achieve in software (including an outboard analog summing unit, see fig. 5). Rather than purchase a wide range of different processors our choices were geared more toward a mastering studio's equipment, limiting equipment purchases to one or at the most two of each needed processor type, opting for those that are generally considered to be superior on acoustic instruments. In the end a significant amount of our outboard gear either used vacuum tubes (valves), or offered the choice between solid state and vacuum tubes signal paths, as in the Millenia NSEQ-2 equalizer. Details appear on our web site.

Figure 5. Control room equipment, Adam (7.1) and Lipinski (stereo) monitors.

2.4 Theatre design, video projection, and screens

The requirements of an ideal space for audio and an ideal space for video are sometimes at odds (see fig. 6). One would ideally like to have very large, very bright images in an immersive environment, and perhaps 3D projection. But consider this: if a 1,000 lumen projector is required to project an image on a screen measuring 2 x 4 meters, a

4,000 lumen projector would be required for the same image and image brightness on a 4 x 8 meter screen, and an 8,000 to 10,000 lumens projector required if the image is in 3D, as 3D glasses block 50-70% of the light (they function by blocking a projector's light intended for the right eye from entering the left eye, and vice versa). The greater the lumen output of a projector, and the greater its resolution, the more power and cooling is required, and thus the greater the fan noise. This is in direct odds with the ideally silent space in which audio can be enjoyed. In movie theaters having only one screen these two requirements can be accomplished with a separate projection room, shooting through glass to the theatre. But in smaller spaces with multiple screens this is not practical. Thus projector brightness and fan noise played a major role in deciding the size of our screens and the power, resolution, and contrast ratio of the projectors; as higher contrast ratios reveal more detail with less light.

Figure 6. Theatre projector and ceiling monitor.

To find the right balance, the entire laboratory was modeled in 3D (see fig. 3), right down to the design of the furniture (using Google's free *Sketch-up*). This was of great assistance in designing the space, trying various sizes, heights, and aspect ratios of the screens, and placements of the projectors and loudspeakers to find a good compromise; screens can't be too high or they will produce neck strain, speakers can't be too low or their sound will be blocked. Using the curved screens and edge-blended projectors often found in virtual reality installations was also modeled, but we determined that the cost of the loss of light, use of glasses, and additional computer for the edge-blending was not worth the added realism. Placing speakers behind the screens was also not a possibility as the room being renovated was small, and the more transparent the acoustics of the screen (the larger its holes) the more light will pass through it, and the brighter the projectors must be. Rather than a virtual reality facility we have opted for a digital art gallery design.

2.5 Quiet please!

As the facility is chiefly concerned with audio, prime consideration was a low noise floor. As stated above, the theatre has a noise floor of 25dB SPL with all equipment off. With all equipment on, including 5 video projectors, 2 computers with extensions to 4 graphic cards, and other equipment, the noise floor raises to only 29dB. Several factors allow this--some obvious, some not.

The most obvious is the discrete HVAC as above, the expense of which is not to be underestimated in the planning stage. The next consideration was fan noise. We found that the noise measurements for video equipment, if available at all, was seldom reliable. Often the vendors themselves did not know the conditions under which the tests were made, including distance from the equipment to the SPL meter, A-weighted or C-weighted, etc. In consequence, fans in almost all the video equipment had to be replaced, re-mounted, and/or refitted with circuitry to slow them down.

2.6 The importance of video screens in lowering the noise floor

A surprisingly crucial component to lowering the noise floor was our choice of screens, for which we initially had given very little thought. The goal was to maintain a sufficiently bright and clear image with a minimum of projector noise, and in an environment that allowed for enough ambient light for informal performances, real-time image processing, certain methods of motion tracking, and face-to-face contact. The optimal answer was in using a very special screen technology, currently patented and used by only one manufacturer, DNP (see fig. 7). Their 'SuperNova' screens are a multi-layered optical element rated for use with the 4k standard and are comprised of 7 layers; some to control black-levels and contrast, others to absorb ambient light from incident angles, and others acting as Fresnel lenses to focus light that would otherwise be reflected into the space (and onto other screens) towards the audience, achieving a screen gain of 2.3 (yes, greater than 1.0). The last layer is very hard plastic which helps protect them from accidental scratches. Aside from allowing a clear image in ambient light conditions, the main benefit is that we were able to use low power projectors, and run them in their low-lamp settings with output of only 350 lumens, which decreases their noise output considerably. The fact that the screen images are front-projected is not always apparent when observing the images, and many who enter the facility are surprised to learn that the images are not from LCD or plasma monitors.

Figure 7. LIATe's DNP screens: 7 layers including Fresnel lenses allowing for a screen gain of 2.3 and rejecting ambient light.

2.7 Loudspeaker placement

There are at the present time no standards for a 24-channel loudspeaker placement. One common placement for 8 speakers is the 'horseshoe', a stereo pair on the front, rear and side walls, which can be duplicated at different levels to form 16-ch, 24-ch, 32-ch, and larger arrays. Unfortunately, this configuration leaves no center channels, which can be approximated with amplitude panning, but often not as convincingly as having an actual speaker in that location (thus the film industry has used center channels from the 1940s onwards, initially at tremendous cost). Researchers from NHK Japan have developed a 22.2 system standard [1] now part of the ITU's Recommendation BS.2051-0 (02/2014). This configuration employs center channels at all height levels and is thus backwards compatible to 2 and 3-channel stereo, 5.1, 7.1, as well as newer configurations used by Auro-3D's of 9.1 through 13.1 configurations, now part of ITU's Recommendation BS.775-3. Thus 22.2 standard was used with slight modifications to meet our given room shape and seating constraints: we have adopted the same 9 ceiling configuration, added 2 more ear-height speakers, and moved the 3 lower speakers to a middle layer between the ear-level and the ceiling (see fig. 8).

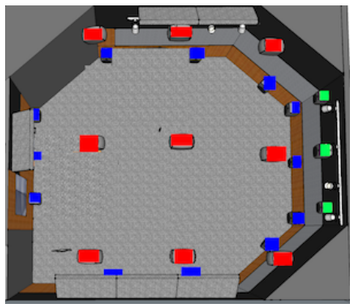


Figure 8. Theatre speaker placement: red are ceiling mounted, blue are at ear level, and green between these levels.

2.8 Theatre equipment design

Monitor choice: Although Adam and Lipinski monitors were chosen for the control room, the Genelec 8000 series monitors (8050 and 8040) were chosen for the 24.2 channel theatre for two primary reasons: 1) their frequency response is quite uniform across models within the series, and 2) they had, by far, the best mounting options. This included 2 meter-long poles to mount the ceiling speakers. In order to achieve the least coloration of the audio system, 3 sets of Metric Halo's ULN-8s were used, as the converters are considered archive quality.

Audio Signal Routing: Although it would certainly be possible to run such a system completely in software, the potentially harmful sound pressure levels that might result from accidental feedback, programing glitches, or equipment freezes meant that having immediate access to a physical device to reduce level controls was a safety concern. The planned use of the facility for real-time DSP, with microphones in the same space as 24 loudspeakers and 2 subs made this all the more important. Digital mixers were considered, but it was found that most are intended for concerts and thus have many more converters for inputs than for outputs; precisely the opposite from our needs. Another consideration is that in order to save space, fader strips are often multi-layered, meaning that one may have to page through the different fader assignment layers before being able to lower the output signal. The use of digital mixers may also contribute to signal degradation that is unavoidable with additional adc/dac conversions. Analog fader boxes were also briefly considered, but we concluded that an analog mixer with direct outs would be the least sonically intrusive solution while providing the most flexible options. Having all the amplifiers at unity gain except the channel faders meant that the timbre of the sounds would endure the least coloration.

One slight disadvantage of most analog mixers is that they usually employ groups of 8 channels, making odd groups of 5, 7, and 9 less intuitive. Although some are specifically designed for 5.1 and 7.1 mixing, these are usually quite expensive. A much simpler solution was found by simply replacing the knobs of the fader channels with knobs of different colors. In our fader system traditional 5.1 and 7.1 formats are color coded as white (fronts) red (center) and green (surrounds). As the ceiling and middle-layer speakers are in pairs of 3, using white or grey colors knobs for the outside speakers of these sets, and red for all center speakers results in groups of 3, 6, and 9 are easily discernable and that the mixer-to-speaker mapping is much more intuitive (see fig. 9). Trivial one might think, but when diagnosing a feedback problem having to search through an ocean of white fader knobs becomes a safety concern.

Figure 9. Theatre's mixer with center channels in red.

Bass management: There are a number of schools of thought about bass management. It is often accomplished by high-pass filtering the bass frequencies for the sub and the rest for the satellites. Another approach championed by John Meyer of Meyer Sound **[ref…]** is to send the full bandwidth signal to the satellites, and to use the subwoofer(s) to compensate for their natural bass roll-off. The analog mixer, allowing all of the input channels to be

summed to the stereo bus easily achieves this latter approach. Bass management can be accomplished quite easily by using direct-outs and aux-sends for outputs to the 24 loudspeakers, and using the stereo bus to feed the two subwoofers.

Computing: The theatre runs on only two linked computers; a quad-core 'i7' iMac runs all audio and control functions, freeing bandwidth so a 6-core Mac Pro can be dedicated exclusively to video processing. The tower's internal PCIe bay has been extended to an external PCIe enclosure (see fig. 10), which houses and supplies sufficient power to additional graphics processors (GPUs). By writing programs that shift the computational load from the CPUs to the four GPUs, LIATe can generate up to 837 million 32-bit pixels per second. This is important because the system has to power 5 HD screens and a control screen, totaling 14 million 32-bit pixels, and running at up to 60 frames per second. Careful use of output dimensions also shifts the task of any video upmixing to the projectors themselves, for which they have dedicated processors leading to excellent results.

Figure 10. Theatre GPUs.

Video signal routing, maxed out:

All of the video equipment has industry-standard RS232 serial ports. The projectors of course come with remotes, but these are less useful with multiple projectors as it is difficult to control a single projector without affecting others. As the facility runs Max software as its primary platform, a patch was made using Max's 'serial' object that controls routing to all of the video equipment's' serial ports (see fig. 11). In this way every aspect of all of the video devices can be easily and simultaneously controlled, and remotely controlled from external devices routed to Max.

Figure 11. Video routing controlled by Max's serial object to video RC232 ports.

2.9 Furniture design for multiple uses

As mentioned in 2.4, the entire laboratory was modeled in 3D, including the furniture. This allowed us to insure that the desk for the audio-tasked computer could be moved to the center of the room for 3D audio research, but also moved backwards to allow for more space when in use as a Theater. The corner pieces to its left folds down to create more space (see fig. 11). The height of the desk and mounting of the computer monitor was carefully planned so that when in use for 3D audio research the screen could be lowered and tilted such while easily seen, it was low enough to allow for a direct path between the monitors and the researcher's ears.

Figure 12. Furniture design for Theatre. Note the desk with the iMac (controlling all audio) is movable to the center of the room for 3D research and to the back of the room for greater seating capacity.

3. RESULTS

3.1 Facility use

The time and effort that went into the design of the facility and its equipment has so far paid off quite well. The control room has excellent acoustics for recording, mixing, and mastering stereo, with or without 2D or 3D images. The Theatre has a multitude of uses. The acoustic isolation, corrective acoustic materials on the walls and ceilings, and removable acoustic screen covers combine to offer professional-level variable acoustics for use as a live room (see fig. 12). It is also ideal for many research and creative activities, informal concerts of electro-acoustic music and intermedia, guest lectures, and presentations. Beyond this, the facility provides a canvass for sound, science and art that captures imaginations, expands horizons, inspires, and informs. It has also become a major attraction for the Music Department and University as a whole.

Figure 13. Recording of Tang Dynasty music for the Qin with covers over the video screens for acoustic balance, 2 Sennheiser MKH 800 twins for the RH and LH of the instrument, and a Neumann U87 for bass frequencies occurring under the instrument and from the instrument stand.

3.2 Output

In the last few years numerous staff and student works created in these facilities were presented in major international festivals and conferences (ICMC, WOCMAT, iKULTURE, etc.), released internationally in cinemas and on major CD and DVD labels (ABLAZE, Ravello-PARMA, Naxos-digital, etc.) and in many local theatrical venues. This includes many works of intermedia and electro-acoustic music from stereo to 124-channels. These include numerous recordings in a variety of formats.

Traditional research has focused on novel algorithms for the upmixing of stereo and 5.1 material to larger speaker arrays and published principally by the Audio Engineering Society. Future plans include work with 16 channel microphone arrays.

Original software has also been a focus, which can be found on our web site[2]. This includes pedagogical software which is now in the process of being packaged into mobile apps.

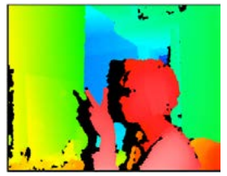


Figure 14. Figure 9-20: The depth map of 'Kinect' is visualized using color gradients from white (near) to blue (far).

3.3 Future plans

Future plans include a greater involvement of researchers and artists outside of Hong Kong. This may take the form of more concerts with calls for works, and providing for residencies for researchers and creative artists to use the facility.

4. REFERENCES

[1] K. Hamasaki, K. Hiyama, R. Okumura, "The 22.2 Multichannel Sound System and Its Application" *Proceedings of the 2005 Audio Engineering Society*, Barcelona, 2005.

[2] http://liate.hkbu.edu.hk/downloads.html

Computer Music as Born-Digital Heritage

Hannah Bosma
Amsterdam
mail@hannahbosma.nl

ABSTRACT

The preservation of electroacoustic and computer music is highly problematic. This is also true for media art and for born-digital cultural heritage in general. This paper discusses the observations, conclusions and recommendations of a Dutch research project on the preservation of born-digital heritage in film, photography, architecture and art that are relevant for electroacoustic and computer music. References are made to research on the preservation of electroacoustic music. OAIS, the ISO standard reference model for digital repository, serves as a starting point. Discussed are the difference between born-digital vs. digitized heritage and the specific concerns regarding born-digital cultural heritage. Attention is paid to the lack of standardization. The recommendations include: to use a distributed OAIS model; to start soon and in an early stage, with simple, basic steps; the importance of education in preservation for art and music students and professionals. The preservation of electro-acoustic and computer music is linked to concerns relating to digital heritage in other cultural-artistic realms.

1. INTRODUCTION

The "digital age" is imbued with the paradoxical problem of preservation and loss. Digital information is copied and kept easily in large quantities. Nevertheless, the preservation of digital information is enormously problematic, due to the vulnerability of storage media, the early obsolescence of hardware and software, the dependence on – and loss of – context and technological environment, complexity of copyrights, the large quantities of information and the lack of selection, ordering and metadata. This is considered a major problem for society and culture: the doom of a "digital dark age" [1, 2].

In electroacoustic music, this issue is very prominent [3–17]. The experimental, innovative, custom-made analogue or digital electronic technologies quickly become obsolete. A lack of standards is caused by technological and musical innovations and by intermingling with such disciplines as theatre or media art. Multiple creative agents are involved, resulting in a dispersion of knowledge and materials.

2014-2015, I investigated the issue of born-digital heritage in film, photography, architecture and (media) art in a research project for the Culturele Coalitie Digitale Duurzaamheid. From my background in electroacoustic and computer music, I will here discuss some of the observations, conclusions and recommendations of this research that are relevant for electroacoustic and computer music.

2. BORN-DIGITAL HERITAGE IN FILM, PHOTOGRAPHY, ARCHITECTURE AND ART

2.1 Background

The Netherlands is investing substantially in a national infrastructure or network for digital heritage [18]. The aim is to promote efficient and effective digital preservation and access for the digital collections of the large variety of archival organizations, libraries and museums in the Netherlands by linking collections, knowledge, and facilities. The work programme on the sustainable preservation of digital heritage is carried by the Nationale Coalitie Digital Duurzaamheid (National Coalition for Digital Preservation, NCDD), a partnership of the National Library, National Archive, National Institute for Sound and Vision (Dutch broadcast archive), Data Archiving and Networked Services (DANS) and the CCDD. Since there is not a single large cultural organization that could represent the whole, diverse cultural field in the Netherlands, several cultural organizations form the Culturele Coalitie Digitale Duurzaamheid (Cultural Coalition for Digital Preservation, CCDD) to represent the cultural sector in the NCDD. The CCDD consists of organizations such as museums and knowledge institutes related to visual arts, cinema, photography, architecture and media art.

Remarkably, the performance arts (music, theatre, dance) are not represented in the CCDD and NCDD. Due to severe financial cuttings in the past years, in the Netherlands there are few specific archival organizations for the performing arts anymore. Because objects are a central concern in the visual arts, collections and archives are apparently considered of more importance for the visual arts than for the performing arts. In governmental prose "cultural heritage" often refers to old buildings, old material objects and visual art only, excluding objects, documents, practices and works of the performing arts.

In the Netherlands, during the past decade there were several large mass digitization projects of national archives. Now that these digitization projects are more or less finished, it is acknowledged that the preservation of born-digital heritage, for which there is no original physical equivalent, poses a different and growing problem.

Various cultural organizations were trying to tackle this on their own. To promote the exchange of knowledge and experiences, the CCDD proposed a research project on generic workflows for the preservation of born-digital heritage in film, photography, architecture and art, the main domains of its members. Moreover, the need was felt to inventory and articulate the requirements of the cultural organizations for the developing national infrastructure for digital heritage. Otherwise, the danger lurked that the cultural organizations would not benefit from it.

2.2 The research project

This research project was called Generieke Workflows Born Digital Erfgoed. It was a 6-month research project of the CCDD, in cooperation with Stichting Digitaal Erfgoed Nederland (DEN), the institute for the preservation of media art LIMA and the Dutch national film institute EYE, financed by the Ministry of Education, Culture and Science (OCW) of the Netherlands. Researchers were Gaby Wijers (director of LIMA) and Hannah Bosma. The research consisted of literature study, in-depth open structured interviews with 34 experts representing 14 Dutch cultural organizations, museums, artists and producers related to film, photography, architecture and art, and an expert meeting (one day, ca. 50 participants). The outcome was a research report and a brochure with conclusions and recommendations, presented at expert meetings and available online for free [19, 20]. The results are incorporated in the work programme on the sustainable preservation of digital heritage of the NCDD.

3. DIGITAL REPOSITORY

A digital repository is an organization that provides storage, preservation and access of digital information, in a secure, reliable way, guaranteeing authenticity and accessibility. It consists of hardware- and software, processes and services, and the required people and means. A standard for digital repositories is the reference model for an Open Archival Information System OAIS, ISO 14721 (fig. 1, [21, 22]). This functional model describes all functions required for a reliable, trusted digital repository. We used the OAIS model as a reference structure for our analysis and for the generic workflow.

The OAIS model assumes that reliable, durable digital preservation consists not only of technically adequate data storage (bit preservation). The data have to be checked regularly. It is required to show that the data were not corrupted by maltreatment or decay (authenticity and fixity information); to transparently ensure the authenticity of the digital objects, the sources and the modifications of the digital objects must be registered (provenance information). Sufficient information on the digital objects is required (metadata, representation information, context information). Access must be provided in usable, permitted formats to the permitted organizations or persons, under the required conditions (e.g., related to copyrights). Monitoring of the technological developments is required to notice whether the stored file formats still can be used (technology watch) and how to prevent their obsoleteness (for example by transformation of the file format). Moreover, the repository itself must be maintained. The organization, including the financial means, must be stable, reliable and durable; in case of discontinuation, the organization must take action to safeguard the collection.

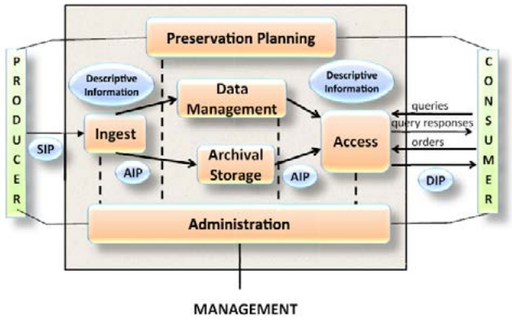


Figure 1. The OAIS model. *SIP* = submission information package; *AIP* = archival information package; *DIP* = dissemination information package.

Standardization of file formats is usually considered as vital for the efficiency and quality of the preservation of large numbers of digital objects in a digital repository. Digital repositories often have strict norms for format standardization or normalization of the ingested digital material.

4. DIGITIZED VS. BORN-DIGITAL

Compared to digitized cultural heritage, born-digital cultural heritage has some specific problems. The problems mentioned below are important for the domains film, photography, architecture and (media) art, but concern computer music as well.

There is **no physical original** or equivalent. This means that the preservation of the born-digital object in its original quality is of essential importance. If the born-digital object gets lost, the object (such as a born-digital art work) itself is lost. There is no physical object that could be digitized again. NB while initially, digitization was proposed as a strategy for preservation, now it is acknowledged that, in general, digital objects are far more vulnerable than physical objects. It is good practice to preserve the original physical objects after digitization. This is of course not possible with born-digital heritage.

Born-digital heritage has far more **variation in software formats** than digitized heritage. While a digitizing organization can choose a convenient, uniform standard format for its collection, born-digital objects have a variety of origins: by different makers in different time periods, in various contexts, with various functions, with and for various equipment, etc. This goes with a variety of software formats and versions. This is even more the case with art and art music (e.g. [10] for electroacoustic music). Whereas mass consuming brings format standardization (to some extent), the digital objects made by artists and composers are often for specialists who present the work to the audience, for example by setting up an installation in an exhibition or by performing a composition in a concert. This goes with a larger variation of software, hardware and file formats. Moreover, cultural heritage institutions may be interested in the intermediate stages of

the creative process of a composer, which adds even more variation.

Large quantities of files are typical for born-digital archives. Since digital storage gets increasingly less expensive and takes few physical space, it is easy for makers to keep many old versions, drafts, intermediate files, etc. For example in digital photography, the number of photographs taken and stored have increased enormously when compared to analogue photography. E-mail is another example, when compared to letters on paper. This is a challenge for institutions that collect artist's archives.

Born-digital objects get **easily lost**. Paradoxically, this goes hand in hand with the above.

a) While it is easy to keep many digital files, it is as easy to throw them away. Some photographers thrash all the pictures of a session after having sent one to the newspaper. Many organizations thrash all e-mail from an employee who leaves the company, while, previously, paper correspondence was filed and stored; this could include correspondence with artists, for example. Thus, historical records and crucial information may disappear.
b) Storage media may become malfunctioning or damaged. Access may be compromised, because of the loss of passwords, lack of permissions, obsolete compression or encryption. Storage media may become obsolete, when they cannot connected to or inserted into newer equipment (e.g., floppy disks, DATs, ZIP-drives, etc.) and the original equipment is not in use anymore.
c) Digital objects may become obsolete, because the old required software (version) is not available anymore and cannot be installed on newer equipment.

b) and c) are common problems in music [13].

A **lack of ordering** is caused by 3) and 4) and by the lack of physical constraints. Different versions and various copies are made easily, kept in different places, storage media and formats. Later, it is often unclear what the authoritative, definitive or original version is of a born-digital object. This goes with a **lack of metadata** and a **lack of context information**. Although there are of course variations, in general physical archives or heritages are better organized than digital ones. Moreover, digitized collections were already selected, ordered and described as the original physical collection, before digitization.

The digital is **not** an **isolated** entity. Born-digital heritage is always embedded in a non-digital context. This includes such diverse aspects as hardware, cultural context, presentation, perception, institutional environments, etc. This is especially the case with complex art works that consist of digital and non-digital elements, like installations or performance works. A one-sided attention to digital preservation alone, neglects the complete works and their contexts. The preservation of digital and non-digital elements has to be integrated.

These problems are present regarding born-digital heritage in general, but are especially prominent in the field of art, music and culture, where authenticity, originality, specificity, creativity and innovation are of essential importance.

5. BORN-DIGITAL CULTURAL HERITAGE

5.1 Standardization and obsolescence

For cultural-artistic born-digital heritage, standardization is often problematic, due to the importance of preserving the originals as much as possible. There is no way to force artists to use only specific standard file formats. Normalization to a standard file format often changes minor or major elements, while with regard to art, this is not allowed or not preferred without consent from the author. For museums, individual art works are of central importance, being of great artistic and often monetary value. In contrast with the use of standardization for the mass preservation by large archival institutions, tailor-made preservation and restoration are the norm for museum's art works, especially when it is for a specific, high-profile occasion such as an exhibition or performance. This also applies to born-digital art works.

The risk of early obsolescence is strongest in the first phases of new technologies, before consolidation, and in niche markets. Thus, experimental art/music and the remains of the creative production process are especially vulnerable.

When it is not possible to perform or install a composition or media art work due to obsolete software and equipment, tailor-made solutions are concerned with emulation, re-construction, repair, re-interpretation, revision or even making a new version, in collaboration with the author and/or other experts [3, 5, 6, 7, 15, 16]. "Authenticity" and "preservation" have different meanings with respect to archival science, art or music [16].

5.2 Domain-specific preservation

Because the technologies and practices of various cultural-artistic domains are so specific, we stress that several aspects of the OAIS model must be domain-specific.[1] This is the case with the *ingest*: domain-specific specialist knowledge is needed for deciding what kind of digital objects, formats, metadata and documentation are welcome or required in the digital repository. The *technology watch* also requires domain-specific specialized knowledge. And it also applies to *access*, related to the specific formats that are used in the domain, the kind of customers or clients, delivery times, delivery methods, copyrights, etc. (for example, while both belong to the audio-visual sector, there are substantial differences between the domains of film and of television with regard to access requirements).

5.3 Beyond OAIS

Very important stages of the preservation process take place outside the OAIS model, at pre-ingest. Already in the production process, decisions and actions take place that are crucial for preservation, such as the choice of software, file formats, equipment and materials, and the documentation of the art work (requirements, technical specifications, intentions of the artist, production process, final result, etc.). Ideally, bit integrity check (by checksum or hash code) would start here and continue through all phases of the preservation workflow. Another important phase at pre-ingest consists of the policy, choices and requirements of cultural organizations regarding the born-digital material to be collected and the consultations and negotiations with makers, producers and suppliers about the selection, qualities, formats, documentation, copyrights etc. of the born-digital objects.

Moreover, subsequent installations, exhibitions, performances and interaction with the audience may be considered to belong to the artwork as well. The documentation of this may be crucial for later re-installation, re-performance, repair, reconstruction or (re)interpretation of the art work. Access is not only, and sometimes never, for "consumers", but often for professionals to exhibit or perform the work; this may feed back into (a revision or addition to) the born-digital object.

We developed a detailed workflow model for cultural organizations that takes these aspects into account: the pre-ingest is more than half of this scheme and there is a feedback line from post-access to pre-ingest.

6. RECOMMENDATIONS

The model of a large, central digital repository that has its own requirements regarding standard formats of digital objects and metadata, is not convenient for born-digital cultural-artistic heritage. Our recommendation is to use the OAIS model not as a scheme for a central digital repository organization, but as a kind of checklist to inventory where and how the various aspects of digital preservation will be organized: a **distributed OAIS model**.

The preservation of born-digital heritage is an on-going process with fast developing knowledge, insights and technologies, both domain-specific and domain-exceeding. Exchanging information and experiences within a specific domain, nationally and internationally, is very valuable; exchanging knowledge with other domains is important as well; this is best dealt with by bundling forces and sharing results in a network, and/or by delegating it to a domain-specific centre. Domain-specific digital preservation knowledge and practices can be developed and distributed via **domain-specific knowledge centres and/or networks**.

But still, there could be an important function for centralized preservation functions. Not with regard to the many domain specific aspects mentioned above, but with regard to **bit preservation**. Bit preservation may seem a simple issue compared to all other preservation problems. However, it is still not a trivial issue at all. Notwithstanding the fast decreasing costs of digital storage, it has become clear that digital storage is expensive, because of the continuing and repetitive costs. Servers cost energy all the time. LTO tapes must be stored in safe, climate-controlled rooms. Digital storage must be checked regularly for bit integrity. Moreover, digital storage must be renewed regularly, because of the expected decay and obsolescence of the storage systems and media. Multiple back-ups must be synchronized. Etc. Cultural-artistic organizations could benefit from central, good, reliable, inexpensive, neutral bit preservation services, while deciding themselves about the file formats and metadata they want to preserve and make available, with the help of the domain-specific knowledge networks/centres that are sensitive to the balance between specific creative/artistic software solutions and the benefits of standardization. Moreover, many artists and other makers have problems with storing their digital work safely and durably (whether because of a lack of knowledge, money or attention). Offering reliable and inexpensive bit preservation to artists and small art/music organizations could be an important step in the preservation chain.

Preservation starts already in the initial composition or production processes of the digital objects. Thus, **education** in the preservation problematic is important as part of the curriculum of art schools and conservatories. Moreover, professional artists, composers and other makers of future born-digital cultural heritage need support and information regarding preservation as well. Such education for both students and professionals should be focused on: 1) raising awareness of the preservation problematic and 2) providing practical solutions.

Although a distributed OAIS model is a good reference, for small organizations and individuals the OAIS model may be too abstract, too ambitious and too unpractical. This may have a paralyzing effect, while it is very important to start with preservation soon. Therefore, some models are developed that provide **first steps for small organizations**, but that are scalable to a reliable digital repository. These start with raising awareness and with inventorying what is already being done regarding preservation. From this follows a next step to take and a next preservation level to aim for. The Levels of Digital Preservation of the USA National Digital Stewardship Alliance (NDSA) is a practical model that advises what a small organization or individual could do [23]. The Dutch *Scoremodel*,[2] developed by DEN and the Belgian digital heritage expertise centre PACKED, is a questionnaire to inventory the strengths and weaknesses of the current preservation practices of an organization and is more focused on organizational and administrative aspects than the NDSA model.

Given the limited resources of most cultural organizations, preservation measures must be as simple and effective as possible. There is a reflex to try to solve preservation problems by developing new software solutions. However, it is important to realize that new software systems often bring new preservation problems. Instead, it might be a good strategy to go **back to basics.** Often, much can be gained by implementing simple, basic standards like adequate file naming, directory organization, sufficient documentation and trying to avoid dependency on proprietary or non-standard software.[3] Checking bit integrity throughout the preservation process requires the freely available MD5 hash checksum utility, organization and discipline. An example of an efficient, low-cost, high-quality domain-specific digital repository is the Dutch institute for media art LIMA, based on LINUX, LTO tapes, much domain-specific and domain-exceeding artistic and technical knowledge, an effective balance between Do It Yourself and collaboration, and a good network.

[1] "Domain" refers to a specific (sub-)section of the cultural sector, such as media art, photography, film, architecture or electroacoustic music.

[2] http://scoremodel.org/

[3] In Dutch, the brochure *Bewaar als...* by Karin van der Heiden offers an overview of such basic preservation standards, see http://bewaarals.nl/.

7. CONCLUSIONS FOR ELECTROACOUSTIC MUSIC

From the perspective of a distributed OAIS digital preservation model, much efforts in the preservation of electroacoustic music are concerned with pre-ingest, post-access and technology watch. We argue that such specialized tasks must be done by or with the help of specialized organizations or individuals and that, on the other hand, centralization could be helpful to provide reliable, low cost bit preservation.

Since decisions and activities in the production process determine later preservation problems to a large extent, preferably preservation must not take place after the fact, but early in the composition and production processes. Therefore, it is recommended to teach preservation awareness, skills and knowledge at art schools and conservatoria and at workshops for artists, composers, performers, sound engineers and other makers. Simple efforts in an early stage could make enormous differences for the preservation later in the future. As such, the author (composer, artist) is responsible for her/his work and its inherent preservation (see also [16]).

The visual arts are organized around the notion of the art object. Music, on the other hand, is concerned with performance. Laurenson pleas to use the notion of the score for the preservation and re-installation of media and installation art [24]. Roeder introduces the notion of performance in the context of digital record preservation [16]. On the other hand, the preservation of electroacoustic music could benefit from the rich theory and experience in the field of media art preservation. A cross fertilization of the preservation practices of media art and electroacoustic music could be fruitful for both domains.

8. REFERENCES

[1] S. Brand, “Escaping The Digital Dark Age.” *Library Journal* 124/2, 1999.

[2] P. Gosh, “Google's Vint Cerf warns of ‘digital Dark Age’”, BBC 13 February 2015, http://www.bbc.com/news/science-environment-31450389.

[3] J. S. Amort, “Case Study 13 Final Report: *Obsessed Again...”* The InterPARES 2 Project, 2007. http://www.interpares.org/display_file.cfm?doc=ip2_cs13_final_report.pdf

[4] M. Battier, “Electroacoustic music studies and the danger of loss,” *Organised Sound* 9/1, pp. 47–53, 2004.

[5] H. Bosma, “Documentation and publication of electroacoustic and multimedia compositions at NEAR: experiences and experiments.” In: Conference Proceedings of the Electroacoustic Music Studies network 05 conference, 2005. http://www.ems-network.org Extended version 2008 in *eContact!* 10.x, Canadian Electroacoustic Community, Concordia University, Montreal, Canada, http://cec.sonus.ca/education/archive/10_x/index.html

[6] H. Bosma, “*Drive* and *Native Tongue*: Intersections of electroacoustic documentation and gender issues.” *Electroacoustic Music Studies network conference 07*, De Montfort University, Leicester, UK, 2007.

[7] H. Bosma, ‘Long live live electronic music: *encore*!’ 9 September 2009, Muziek Centrum Nederland, Amsterdam.

[8] S. Canazza and A. Vidolin, “Introduction: Preserving Electroacoustic Music,” *Journal of New Music Research* 30/4, pp. 289-293, 2001.

[9] A. P. Cuervo, “Ephemeral Music: Electroacoustic Music Collections in the United States,” *Research Forum Peer-Reviewed Research Papers* pp. 1-6, 2008. http://dx.doi.org/doi:10.7282/T3KH0Q1P

[10] J. Douglas, “InterPARES 2 Project - General Study 03 Final Report: Preserving Interactive Digital Music - The MUSTICA Initiative,” 2007. http://www.interpares.org/display_file.cfm?doc=ip2_gs03_final_report.pdf

[11] S. Emmerson, “In what form can live electronic music live on?” *Organised Sound* 11/3: pp. 209–219, 2006.

[12] M. Guercio, J. Barthélemy and A. Bonardi, “Authenticity Issue in Performing Arts using Live Electronics,” *Proceedings 4th Sound and Music Computing Conference (SMC 07)*, Lefkada (Greece), 11-13 July 2007.

[13] M. Longton, “InterPARES 2 Project - General Study 04 Final Report: Recordkeeping Practices of Composers,” 2007. http://www.interpares.org/display_file.cfm?doc=ip2_gs04_final_report.pdf

[14] R. Polfreman, D. Sheppard and I. Dearden, “Time to re-wire? Problems and strategies for the maintenance of live electronics,” *Organised Sound* 11/3: pp. 229-242, 2006.

[15] J. Roeder, “Preserving Authentic Interactive Digital Artworks: Case Studies from the InterPARES Project.” *International Cultural Heritage Informatics Meeting: Proceedings from ICHIM 04, Berlin, Germany, 30 August-2 September 2004*, Xavier Perrot (ed.), Toronto: Archives & Museum Informatics, 2004.

[16] J. Roeder, “Art and Digital Records: Paradoxes and Problems of Preservation,” *Archivaria* 65: pp. 151-163, 2008.

[17] D. B. Wetzel, “A Model for the Conservation of Interactive Electroacoustic Repertoire: Analysis, Reconstruction, and Performance in the Face of Technological Obsolescence,” *Organised Sound* 11/3: pp. 255–272, 2006.

[18] B. Sierman and M. Ras, “Best until… A national infrastructure for Digital Preservation in the Netherlands,” *12th International Conference on Digital Preservation*, University of North Carolina at Chapel Hill, 2015. http://digitalpreservation.nl/seeds/wp-content/uploads/2015/01/ipres2015_sierman_ras.pdf

[19] G. Wijers and H. Bosma, *Generieke Workflows Born Digital Erfgoed. Behoud van Born Digital Erfgoed in Nederland: Film, fotografie, architectuur, kunst.* Stichting Digitaal Erfgoed Nederland, 2015. http://www.den.nl/art/uploads/files/20150622_CCDD-BornDigitalOnderzoek-def.pdf

[20] G. Wijers and H.Bosma, *Born digital cultureel erfgoed is bedreigd erfgoed: Op weg naar een generieke workflow voor born digital erfgoed binnen de domeinen kunst, film, fotografie en architectuur.* Culturele Coalitie Digitale Duurzaamheid. http://www.den.nl/art/uploads/files/Publicaties/Born_Digital_erfgoed_is_bedreigd_erfgoed.pdf

[21] Consultative Committee for Space Data Systems, “Reference model for an Open Archival Information System (OAIS),” Recommended practice CCSDS 650.0-m-2 Magenta Book. Issue 2. Washington, D.C.: CCSDS, June 2012. http://public.ccsds.org/publications/archive/650x0m2.pdf

[22] B. Sierman (2012) “Het OAIS-model, een leidraad voor duurzame toegankelijkheid.” *Handboek Informatiewetenschap*, december 2012, part IV B 690-1, pp. 1-27. https://www.kb.nl/sites/default/files/docs/sierman_oiasmodelned.pdf

[23] M. Phillips, J. Bailey, A. Goethals and T. Owens, “The NDSA Levels of Digital Preservation: An Explanation and Uses.,” 2013. http://www.digitalpreservation.gov/ndsa/activities/levels.html

[24] Laurenson, Pip. “Authenticity, change and loss in the conservation of time-based media installations.” In: Judith Schachter and Stephen Brockmann (eds.). *(Im)permanence: Cultures in/out of Time*. The Center for the Arts in Society, Carnegie Mellon University, 2008.

COMPUTER MUSIC INTERPRETATION IN PRACTICE

Serge Lemouton
IRCAM-CGP
serge.lemouton@ircam.fr

ABSTRACT

Computer music designer is still a new job, emerging as a professional practice only in the last decades. This function has many aspects; personally, I consider that one of the most important, and not well-documented parts of our job is the concert performance. In this paper, I will discuss this discipline (performing live electronic music) from a practical point of view. I will illustrate this idea with short presentations about the interpretation of some existing classic pieces of the electroacoustic mixed works repertoire.

1. INTRODUCTION

The development of mechanical music technologies (recording, analog and digital techniques, etc.) has had consequences and raised some questions about musical activity and about the category of musical interpretation: can we speak of “music” without interpretation? What is the status of the recording of a piece (between the score and the concert)? Now that we have audio recordings of the entire musical repertoire, why should we still build concert halls?

Since the beginning of the 20th century, composers (such as Stravinsky, Ravel, Bartok, etc.) foresaw the consequences of sound recording technologies on the musical interpretation of their works. And of course, the influence of sound technologies on musical composition continued to increase during the last century, from analog techniques to our current digital world.

In this paper I will focus on the category of musical interpretation. First of all, interpretation is different from performance. Interpreting is more than performing, playing computer music is not only performing it, but has many more aspects.

At IRCAM, this activity is now taught in a special workshop “AIRE” (Atelier Interpretation des musiques électroacoustiques), held during IRCAM’s ManiFeste Academy since 2012.

More and more often, the person identified as a “technician” or a “sound engineer”, became integrated in, appointed by, and toured with a number of new music ensembles involved in the performance of mixed music. For example, we can cite John Whiting in the vocal ensemble Electric Phoenix or Scott Frazer[1] with the Kronos quartet. More and more frequently, this function is recognized not only as a technical role but also as musicianship.

2. WHY?

Why is interpretation necessary for electroacoustic works, whether they belong to the “real-time” or the “tape music” category?

2.1 Live music

Most of the time, when we are speaking about “Music” today, we are speaking about recorded music, about music reproduction—dead music. Music, to be alive, should be performed “live”; an audio recording is simply a trace of a musical event. But we always speak about it, quite improperly, as music. As early as 1937, Bela Bartok was aware of the danger of what he called “mechanical music” vs. the variability of live music.

But what is the status of purely synthetic music? Is it too conservative to consider that music that is not performed is not music?

2.2 Interpretation against obsolescence

Interpretation is a way to overcome the technological obsolescence that every computer musician knows very well. The obsolescence of the technologies used by musicians in real-time works can be seen as a danger, as a risk for the existence of these new forms of musical expression [1].

It is possible to compare scores written on paper with a lifespan that can be measured in centuries – we can still find music written down in the Middle Ages – with digital supports whose instability can be measured daily, at our expense. But an antique parchment only has value to the person who knows how to read it, that written music remains virtual if it is not sung.

In the beginning of IRCAM, no one was aware of the seriousness of the problem: the works produced in the 1980s were made with a total lack of concern for this issue or with an optimistic technophily. We realized the problem later, in the beginning of the 21st century.

IRCAM is now concerned by the conservation of the works created in its studios. To create its repertoire, the institute asked composers to write works interacting with the institute’s research departments. This concern for the conservation takes the form of archives on different supports and documentation written by tutors/assistants/computer music designers. Valorizing the

[1] http://www.allmusic.com/artist/scott-fraser-mn0001479637/credits

works by performing them in concert and on tour leads to the creation of an original repertoire. The conservation of this repertoire is obviously a part of the will to create a history, a kind of tradition.
The experience of computer music designers, who must transfer sometimes complex works to perform them again (at IRCAM we call this action "porting") from one system to another as technology evolves from one generation to another (from the historical 4X to the IRCAM computer music station and different versions of MAX software), has led us to invent, to develop, a specific *savoir-faire* of the techniques and practices that have made it possible to save almost the entire catalogue of works created at IRCAM (more than 700 works) from digital ruin.[2][3]

2.3 Interpretation as renovation

Moreover, porting a musical work to a new technology is not only a way to overcome technological obsolescence, but also esthetical aging. It seems especially true for tape music; we often have this impression while listening to old recordings that they sound "dated".

2.4 Interpretation and notation

The score is an integral part of our serious art music ("musique savante"). Even if all music is ephemeral and immaterial, the act of writing it down inscribes it in history and in the effort of "the desire to last". Not all composers seem particularly worried about the future of their works; creation is more about renewal, about a flow, than about keeping, storing, archiving. And yet, if composers write down their music, it is for its survival. The score is both a way to transmit music to the performers and a support that enables its long-term preservation. In this respect, electro-acoustic music, and particularly interactive mixed works, creates numerous problems because today there is no universally shared musical notation.
The conservation of electro-acoustic works seems impossible without the performers. The computer music designers are both archeologists of a near past, specialists of obsolete technologies, interpreters of musical texts, and virtuosos of new musical technologies. The responsibility of transmitting the composer's will with authenticity lies with them.

2.5 Interpretation and transmission

In the domain of mixed music, scores are very often incomplete. Consequently, the only possible transmission of these highly technological artifacts still relies heavily on oral tradition!
We can suppose that composers should play their own music. But is the composer the best interpreter of his music[2]? And if not, why? Some of them are really expert in the art of sound diffusion but not all of them. As composers spent a lot of time listening to their own sounds in the studio, they don't necessarily have the same perception of it as the listeners in the concert hall. And unfortunately, composers are human, and consequently mortal.
In the real-time music context, I often have to face this paradoxical situation: real-time should come with the acceptation of the unexpected. But very often, composers are not ready to accept the unexpected in their music.
"The term "real-time" in musical composition can be inaccurate because a part of the musical components is often predetermined, and is not subject to variation from an interpretation to the other." [5]

[2] "Le compositeur n'est sans doute pas toujours le mieux placé pour interpréter ces propres œuvres, même si cette solution prévaut aujourd'hui (en l'absence d'un nombre suffisant d'interprètes reconnus et en raison, entre autre, du surcoût financier que cela occasionne)" [4, note 60].

3. WHAT?

3.1 What is musical interpretation?

As mentioned, interpretation is more than performance: it is a complex activity. In the classical music context, a musical interpretation requires the ability to read the music (knowing the vocabulary) and to understand the text (knowing the syntax). It also means mastering its instrument (it takes years of practice to make a virtuoso), interpreting the composer's will (knowing the stylistic context). Finally, the musician should be able to perform the music in concert, interacting with the audience, the hall, and the other performers.

3.2 Can we speak of musical interpretation for computer music?

For computer music, things are slightly different because of the nature of the "instrument". There is an extra step: constructing the instrument. In this sense the computer music performer is also his own instrument-builder (luthier). Moreover, there is no school or conservatory to learn how to become a computer virtuoso today.

3.3 Interpretation and "real-time" music

To allow the possibility of an interpretation, in every sense of the word, there must be a text to be interpreted. An exegesis is only possible if the following elements are present: a text, a tradition, and an interpreter.
What could be the meaning of interpretation in the context of what is called "real-time music"? In electroacoustic music, the text is almost always missing. The notion of tradition is also problematic because "real-time electronic music" has a relatively short history of about 40 years: it is a young tradition, but it exists.
Real-time has always been presented as a way to reinstate the function of the instrumentalist and his instrument is the electronic music context:
"The main advantage of real-time systems is the following: with them, the player is no longer a slave to the machine. For this purpose, the machine has had to become more intelligent, or at least, to simulate a part of the musician's activity in performance situation."[6]
The real-time concept is a result of technological evolution and also a historic process dating from the first tape music pieces from the 1950s, through the mixed music practices of the 1960s, ending with the real-time music repertoire. The practice of mixed music was an answer to the lack of musical instruments in tape-only music. Real-time was an answer to the lack of interpretation in mixed music works.
The topics of musical interpretation and real-time technologies are obviously strongly interwoven. I was able to observe the relations of these concepts during my personal experience of more than twenty years of real-time music at IRCAM.

3.4 The computer as a musical instrument?

Real-time synthesis allows us to use the computer as a musical instrument. The computer can be used in a concert situation, "played" by a musician. But it is a peculiar kind of instrument because it doesn't possess a specific shape.
In computer music, controlling the computer as a "virtual instrument" is related to the development of gestural controls for electronic devices. It is also a question of synthesis control. Current acoustic synthesis techniques can surpass the musical instrument's limits, but at the same time, the musician control is still extremely simple, limited (very often to keyboard- or sliders-type), and rudimentary compared to the expert interaction involved between the virtuoso and his/her instrument.

3.5 Interpretation of mechanical music

Musical interpretation on a instrument involves playing with the instrument give. In English also it is possible to play on the meaning of "play": the ludic (a score is like the rules of a game) and the mechanical (an instrument has numerous mechanical degrees of freedoms, "avoir du jeu")
Rubato, swing are freedoms that the musician can take with the chronometric curse of time. But designing a machine able to produce a convincing swing is quite a serious challenge in artificial intelligence!
Real-time permits true interaction between musician and machine, i.e. reciprocity, a dialog going in both directions, similar to what happens between musicians playing together.

3.6 Interpreting space

We have seen that real-time allows the reintroduction of traditional characteristics of musical interpretation, by the flexibility that these techniques bring, compared to prerecorded fixed sounds. But it also brings into play a new kind of musical interpretation in the spatial dimension of sound diffusion and sound projection. A very important and new domain of electroacoustic interpretation is spatial diffusion.
A new kind of instrumental practice emerges: spatial interpretation is a prolongation of concrete and electronic tape music practices. This role is often (but not always) undertaken by the composer. During the concert, the electronic sounds are projected into the concert hall space using the mixing desk faders or specific electronic devices. It can be a simple fixed assignment of the audio tracks to specific loudspeakers or spatial trajectories of sound sources controlled by either manual or automatic processes. The loudspeaker setup can be frontal, surrounding the audience on a horizontal plane, or even in a three-dimensional sphere around the audience. Space has become a compositional parameter that should be interpreted and performed live, in function of the music style and in function of the concert hall acoustics, dimensions, and configurations.

3.7 Obsolescence and re-interpretation

As mentioned earlier, real-time musical works evolve with time. Technological evolutions imply these works are a kind of life form that depend heavily on these technologies. Real-time music works should perpetually be adapted or die. Porting, re-mixing are also new forms of re-interpretation of the will of the composer by this new kind of interpreters that today are called "computer music designers" or computer musician.

3.8 New species of musicians

At the end the computer (and the sound recording technologies) hasn't replaced the real life performance of living interpreters. On the contrary, it has demonstrated the crucial importance of humans in music, and brings to life new interpretative practices, new disciplines, such as acousmatic music sound diffusion, turntablism, DJs, or computer music designers.

4. HOW? : INTERPRETATION IN PRACTICE

In this last section, I will present some real cases of musical pieces from the classic electro-acoustic repertoire, from the point of view of the computer musician. Because interpretation is not only knowledge and skills, but mainly a practice, the only way to know how to perform these pieces is by rehearsing and playing in concert. The examples and anecdotes presented here are taken from my experience and repertoire as a computer music designer.

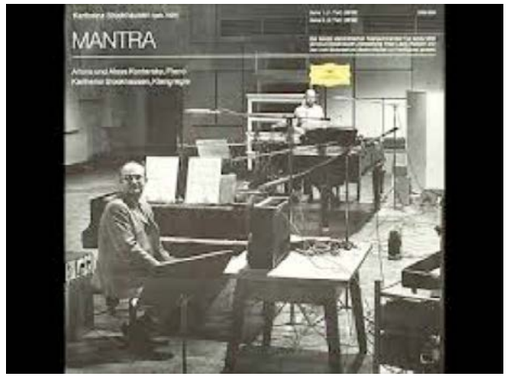
4.1 Luigi Nono

"Electronic sound transformation, timbral distribution and time spaces does not mean the rigidity of the electronically extended sound, but the personal interpretation, a very important point for Nono." (Hans-Peter Haller, *Diary note 3.9.84*).
As a consequence of this esthetic, Luigi Nono's music can only be played by people to whom he transmits the knowledge such as Andre Richard (who defined himself as "a composer, conductor, and performer of live electronic music") or Hans Peter Haller. It illustrates the oral tradition nature of the live electronics repertoire. Some modern technical re-interpretations are documented in [7].

4.2 Stockhausen: Mantra

In 1970, the original version of Mantra required some analog gear: sine wave generators, shortwave radio receivers, and ring modulator. These devices are integrated

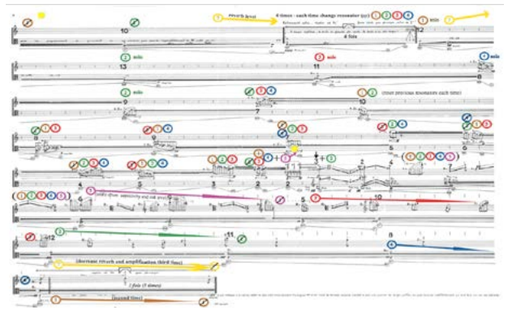
in the instrumentarium played by the two pianists, in what is often considered as the first important piece of the "live transformed" repertoire.



In the score, the composer precisely describes the characteristics of the required hardware. But as analog equipment of this kind was getting more and more difficult to find in the beginning of our century, Jan Panis realized the first digital version. Miller Puckette also wrote a computerized version of *Mantra* in his "pure data repertoire" [8] (http://msp.ucsd.edu/pdrp/latest/files/doc/)
Even if one finds a shortwave receiver, the Morse code that could still be heard on these frequencies in the 1970s have vanished today, so they are replaced by a recording. This is not without consequences on the philosophical esthetic of the piece! The consequence of the evolution of the available controllers and electroacoustic devices is that each time such a piece is performed, new realizations are necessary.

4.3 Grisey: Prologue

« All the works I have written using electronics have had to be constantly reviewed because of technological evolutions. If you write a piece for electronics, you should always renew the system to make it available to the concert hall. Technology forces me to look back and to work again. A new kind of tape. Going from tape to computer. And from a computer to a new computer model. Or from a synthesizer to a new synthesizer model. It has no end.» [9]
Prologue is the viola solo opening his *Espaces Acoustiques*. If played alone, it should be played through five acoustic resonators (a snare drum, Ondes Martenot "diffuseurs", a tam-tam, etc.). In 2001, Eric Daubresse realized a computerized version virtualizing the resonators.
The performance of the electronic part is rather virtuosic; the level of the viola sound exciting each resonator has to be controlled as written in the score:



The level of six faders on the mixing desk should be moved simultaneously, sometimes very quickly and precisely; it requires several rehearsals with the soloist to be able to perform it comfortably.

4.4 Manoury: Jupiter

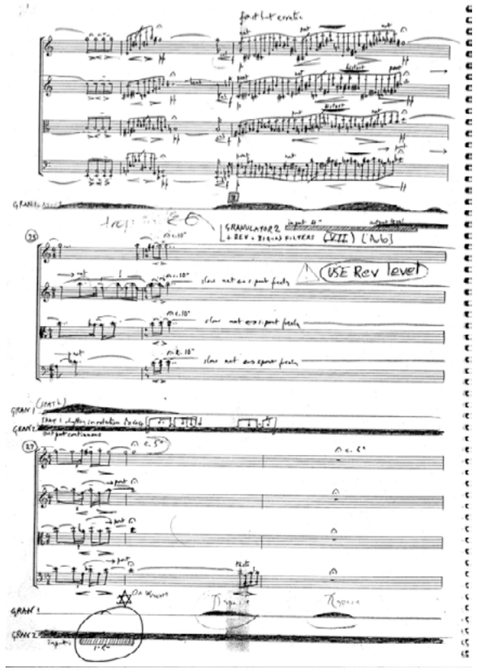
This historic piece is a seminal work in the "real-time music" repertoire. It happens to be played quite often since is premiere and it is certainly very interesting to consider that it is probably the piece that had the most hardware and software implementations.
For real-time piece it is very important that different instrumentalists perform it. Before having played it with several young flutists in the *Centre Acanthes* Academy in 2000, I had not realized the variability of the electronic part of this piece that I always played before with the same very virtuosic but predictable flute player. As the sound of the flute was always the same, the electronic part sounded identical (not so different from a tape) but when I happen to be confronted by other flute sonorities and interpretations, the listeners were able to feel that the computer was reacting in real-time.
Jupiter has had a lot of different technological implementations, from the first version using the 1987 experimental cutting edge IRCAM technology (the 4X) to the present day versions. It was ported to at least five different hardware platforms and five software versions: this is certainly a record!

1987: 4X
1992: NEXT
1997: SGI
2001: MAX/MSP
2003: PureData
2015: Faust, Web Audio ?

We can ask ourselves the question of authenticity: which version is the authentic? Is it the first one, or the last one? We can assess more certainly that they have all some kind of authenticity!

4.5 Harvey: *Fourth Quartet*

Jonathan Harvey was also concerned by the interpretation of the electronic part of his works. Since *One Evening* (1993), he requires the presence of two technicians to perform the electronic part.
I had the opportunity to play the electroacoustic parts in concert of *Madonna of Winter and Spring* (1986, requiring a total of 5 *operators*), *Soleil Noir/Chitra* (1994, 2 operators), and *Bird concerto with piano song* (2001, *sound "diffusionists"*) alongside the composer. He was very precise about the kind of effects required, but he was also always insistent on the musicality of the interpretation of these effects. It was always a very nice experience.
In the *Fourth String Quartet* (realized in collaboration with Gilbert Nouno), which represents a kind of achievement of all his previous experiences on the integration of electroacoustic media in his musical language, a lot of importance is given to the spatial diffusion of the electroacoustic transformations that can be freely drawn in space using a drawing tablet.



5. CONCLUSION

It can't be denied that electronic or computer music become a logical or natural part of contemporary music; and the integration of technologies in this universal art form is not without consequences on music practices. But for all that, it does not mean that music can be distributed directly to the listener. On the contrary, I have shown that computer music has created the need for new specialized musicians, not composers, not instrumentalists, but "interpreters", because any kind of music cannot live without an audience, in front of which music should be performed.

6. REFERENCES

[1] Bernardini, N. & Vidolin, A., "Sustainable live electro-acoustic music". In *Proceedings of Sound and Music Computing*, Salerno, Italy, 2005.

[2] Bonardi, Alain and Barthélemy, Jérome, *The preservation, emulation, migration, and virtualization of live electronics for performing arts: An overview of musical and technical issues*. Journal on Computing and Cultural Heritage (JOCCH 1, 1: 6–16), 2008.

[3] Lemouton, S. & Goldszmidt, S, "La préservation des œuvres du répertoire IRCAM : Présentation du modèle Sidney et analyse des dispositifs temps réel". In *Journées d'Informatique musicale*, Albi, 2016.

[4] Tiffon, Vincent, *L'interprétation des enregistrements et l'enregistrement des interprétations : une approche médiologique* (http://demeter.revue.univ-lille3.fr/interpretation/tiffon.pdf).

[5] Manoury, P., *Considérations (toujours actuelles) sur l'état de la musique en temps réel*, 2007 (http://www.philippemanoury.com/?p=319).

[6] Manoury, P., *De l'incidence des systèmes en temps reel sur la création musicale*, 1988.

[7] Polfreman, R., Sheppard, D., & Dearden, I., *Re-Wired: Reworking 20th century live-electronics for today*. In Proceedings of the International Computer Music Conference, Barcelona, Spain, 2005 (pp. 41-44).

[8] Puckette, Miller, *New Public-Domain Realizations of Standard Pieces for Instruments and Live Electronics*. Proc. ICMC, 2001.

[9] Grisey, Gérard, "Entretien avec David Bündler" (1996), in Gérard Grisey, *Écrits, ou l''invention de la musique spectrale*, Paris, Éditions MF, 2008.

New Roles for Computer Musicians in S.T.E.A.M.

Jeremy C. Baguyos
University of Nebraska (Omaha)
jbaguyos@unomaha.edu

ABSTRACT

Computation has long had a role in the musical arts, especially in musical composition. But how has music and the creative process informed the Information Sciences? What more can it do to enhance/inform/improve/innovate the information sciences (both theoretical and practical, science and business). To answer these questions, this paper gives an overview of the role of the creative process, esp. the role of music's creative process, in the information science and technology disciplines.

The S.T.E.A.M. (Science Technology Engineering Arts Mathematics) initiative has renewed the interest in the role of the arts in information science, and as part of that effort, this paper examines the direct role of a formally trained computer musician functioning as a faculty member within an information science and technology academic unit.

1. INTRODUCTION

Much has been written about the importance of the creative arts within information science and technology disciplines. The following are three representative examples.

1.1 President's Committee on the Arts and the Humanities

Created in 1982 under President Reagan, the President's Committee on the Arts and the Humanities (PCAH) is an advisory committee to the White House on cultural issues and stated that "Policymakers and civic and business leaders, as reflected in several recent high level task force reports, are increasingly recognizing the potential role of the arts in spurring innovation."-----*President's Committee on the Arts and the Humanities, 2011*

1.2 p21 Survey

Decades after their findings, a p21 survey of employers about their views of the preparation of college graduates in innovation and creativity indicated that much improvement can still be pursued in the area of creativity and innovation.

"Creativity/Innovation is projected to 'increase in importance' for future workforce entrants, according to more than 70 percent (73.6 percent) of employer respondents. Currently, however, more than half of employer respondents (54.2 percent) report new workforce entrants with a high school diploma to be 'deficient' in this skill set."-----*Are They Really Ready to Work? Employer's Perspectives on the Basic Knowledge and Applied Skills of New Entrants to the 21st Century U.S. work Force, 2006*

1.3. "The Creativity Crisis"

Despite the general association of creativity and innovation with success, creativity remains tenuously connected to the information science curriculums and little practical literature has been generated.

"Creativity has always been prized in American society, but it's never really been understood. While our creativity scores decline unchecked, the current national strategy for creativity consists of little more than praying for a Greek muse to drop by our houses."-----*"The Creativity Crisis,"* Newsweek*, 2010*

In addition to contributing to a relatively new research area, this S.T.E.A.M. (Science Technology Engineering Arts Mathematics) paper elaborates on the role of the arts within STEM (Science Technology Engineering Mathematics) fields, the enhancement of STEM fields, and pursues the cogent and seamless integration of creativity as it pertains to the information science curriculum. Conversely, while studying cross-disciplinary applications, musicians can reassess, reaffirm, enhance/improve, and innovate the creative process itself, as well as doing the same for information science.

2. THE MUSIC FACULTY'S CURRICULUM WITHIN INFO SCIENCE

This section outlines what is taught by a music faculty member (trained and degreed in music) within an information science curriculum at the University of Nebraska (Omaha). Most topics are taught, have been taught, and are central to introductory undergraduate courses in Information Technology Innovation, Management Information Systems, Computer Science, and Multimedia. Most of the students in the classes are majoring in computer science, management information systems, IT Innovation, engineering, business administration, marketing, entrepreneurship, music, and the arts. What follows are not lesson plans (that would be too large a task for this paper). Instead these are a re-introduction of topics central to the arts and music, but re-formatted/re-purposed for an information science curriculum.

2.1 Incubation of Creativity, Ideation, Innovation, and Empathy

"Ideation" is a term used to describe where most ideas begin in the iterative design process of a new product or service in the technology sector. A creative individual sees a "need" (artistic or functional) and comes up with an idea to solve the "need." In order for the process to begin, it is necessary to initialize the conversation immediately with potential users of the solution, whether artistic and/or functional (Ries 2011). A successful ideation process depends on this conversation. So an initial "need" and "solution" needs to be identified, for purposes of starting the discourse around the design process. Although there will be much pivoting later in the process, we have to start with something (a theme), even if it is to iterate/fork/vary the original idea (like variations of the theme) or to reject the original idea, altogether, for another idea (another theme). The difference between the innovation sector and the music sector in this iterative theme and variations approach is the musical artist must place more consideration on the conversations with potential users of an idea, when applying a theme and variations approach to information technology innovation/entrepreneurship.

"Incubation" is a term used to describe and explain optimal ideation environments, so that one is most creative and aware. One of the most important principles is to assure that the imagination is unconstrained by negative thoughts, physical exhaustion, distracting worries, stifling daily routines, and rigid mindsets. "Incubation" also involves activities that assure that the mind is focused, has a heightened, conscious awareness of surrounding details and how details fit into a larger system of interacting details, is always looking for patterns and interdisciplinary connections between seemingly unconnected ideas, is informed by pertinent new information and ideas in addition to the traditions that shape an artist, is pushed to recognize and imagine new possibilities for existing structures, and most importantly, is pushed to meet and document the expectations of incubation and ideation on a daily basis--sometimes expressed in an idea journal, similar to a writer's journal (Michalko 2006).

Other incubation practices involve John Cage. Cage is historically intertwined with "ideation" and "mind pumping" creative practices. Those practices are summarized elegantly in a list of ten paradoxically named "rules," although the rules are misattributed to John Cage. They should be correctly attributed to Sister Corita Kent at the Department of Art at Immaculate Heart College. Some of the more pertinent "rules" that can be applied to "mind pumping" within information technology and innovation include the following:

1) Find a place that you trust and then try trusting it for a while,
2) consider everything an experiment,
3) nothing is a mistake, there's no win and no fail; there is only make,
4) the only rule is work,
5) don't try to create and analyze at the same time; they're different processes,
6) be happy whenever you can manage it,
7) there should be new rules next week (Kent 2008).

Although authored by Corita Kent, the "rules" have made their way to computer musicians by association with John Cage. Many of the "rules," all rooted in the sensibilities of musicians and artists, are echoed in contemporary business and technology publications about innovation and creativity (Michalko 2006).

Musical artists are good at getting to core meaning, laddering down from functional feature sets to why those feature sets are viscerally desired in the first place. The history of the music literature is grounded in emotion (like the 19th century) as well as functional analysis. A musical artist's day-to-day job rests with the mediation of an auditory message with an audience, on mostly subjective and emotional terms, regardless of the musician's implementation of said auditory structures using the constructs of the musical language. A composer may be a master of enharmonic modulations, but most of the audience will only care about the enharmonic modulation's emotional content, however the audience may define it. Likewise, the development of information technology solutions is grounded in the emotional resonance of a solution, as well as its function and features upon which the emotional resonance is built. It is the musical artist's inherent empathy that must be tapped. Features and functionality and even usability are only table stakes prerequisites for the success of an information technology solution. The true success of an idea rests with its emotional resonance, the last place of abundant opportunity since functionality and features can be cheaply researched and implemented. This leaves as the final frontier of opportunity, emotion and empathy, which is naturally part of the core sensibilities invoked by musical artists in their day-to-day work (Pink 2006).

Towards designing for emotional response, the app must elicit some kind of response as a starting point. A non-response is probably an early warning of an unsuccessful idea. Good ideas elicit a response without prompting, even if the responses are polarized (Kawasaki 2015). Then, the response must be accurately identified in order to collect accurate insight for the ideation and design process (Maddock 2011). Usability attributes that resonate emotionally include whimsical, organic, glowing, imbued with personality, or any emotional descriptor that speaks to individuation and pleasurability. Have an aesthetic and never be satisfied with merely "functional" (Wang 2014).

Laddering is a classic technique, utilizing a simple but relentless series of "why" questions, that methodically identifies the emotional resonance of an idea, a piece of art or music, a product, a service, or anything else. Laddering is a technique that has been used by design professionals, but also by musicians, artists, and writers (alt-

hough not necessarily with the same terminology and profit motives). Whether an entrepreneur and/or a musical artist, laddering identifies the emotional component of and the emotional connection to product ideas that were conceived strictly out of what was thought to be only functional aspects, and to art and music that were conceptualized outside of its potential impact on an audience. More importantly, laddering helps provide a clear, truthful picture of the "pain" for which an idea, a potential idea, or future idea is trying to solve (Maddock 2011).

It can be assumed that the pursuit of the more subjective aspects of design, such as individuation and pleasure, do not displace the underlying importance of safety, functionality, and usability. Emotional aspects of individuation and pleasure cannot happen without the successful fulfillment of safety, functional, and usability aspects. (Pink 2006).

2.2. Ideating with S.C.A.M.P.E.R.

After field work and research about an initial idea (using laddering or some other technique), the next step is rapid ideation. In this highly creative process, whether accomplished by groups or an individual, the object is to quickly generate as many ideas as possible that could potentially solve an identified "pain." After a large pool of ideas is generated, they should be evaluated for the best ideas.

Here is a popular ideating technique for solving the identified "pain." There are many ideation techniques, but the one selected for this article has been chosen because of its widespread use, its comprehensive approaches, and the similarity to approaches used by creative artists. A type of lateral thinking ideation technique, and similar to a theme and variations, is the S.C.A.M.P.E.R. technique. The general category of lateral thinking exercises involves methodical and organized examination of a problem from multiple viewpoints. S.C.A.M.P.E.R. takes an existing idea—or theme (whether old or very new), and methodically manipulates and modifies the idea into something new—similar to musical variations). Like many ideation techniques or composition techniques, the process is highly iterative.

There are other lateral thinking techniques, but S.C.A.M.P.E.R. is one of the most popular and incorporates many other separate lateral thinking ideation techniques. It is almost a suite of best practices in lateral thinking ideation. The letters in S.C.A.M.P.E.R. refer to an action that you take upon the idea or an aspect or attribute of the idea.

S = **S**ubstitute something
C = **C**ombine it with something else
A = **A**dapt something to it
M = **M**odify or **M**agnify it
P = **P**ut to some other use
E = **E**liminate something
R = **R**everse or **R**earrange

S.C.A.M.P.E.R. involves some simple "theme and variation" ideation approaches, where an attribute is subtly modified (although the outcomes are not necessarily subtle). These are S "substitute something," M "Modify or Magnify," or E "eliminate something." A "adapt" refers to how someone else's "theme" can be adapted into or "quoted" in the variation of the theme. Although S.C.A.M.P.E.R. is classified as a lateral thinking technique, there are provisions for free association techniques and synthesis techniques (not in the computer music community's use of the word synthesis, but rather in the creative community's use of the word). Synthesis involves cogently combining two elements that otherwise do not seem to belong together. In this forced combination, a new idea or attribute of an idea is generated. Obviously, C, "combining" is a synthesis technique. P, "put to some other use" is also a form of synthesis in that one is combining an existing idea or attribute with a new purpose. C and P are acts of orchestration. Two separate ideas or attributes are combined to create a new idea or attribute. This technique can be a very powerful technique. Although ubiquitous and mundane today, somebody had to engage in the creative act of imagining the mobile phone and the Internet combining to create a smartphone. R "reverse" is a gateway to free association creativity techniques such as the uncontrolled free association technique of mind mapping (called "think bubbles" in some literature). Mind maps allow for unconstrained free association with a concept and the unconstrained free associations with the newly generated concepts, resulting in a spider web of quickly generated ideas that can be thematically grouped. Mind mapping is a technique that is often used in group brainstorming. The R "reverse" is also related to mind mapping in that one is supposed to reverse any assumption they have about a concept, idea, or attribute. For example, phones are not for calling people. Or, music does not have to make sound. Clearly, the artistic John Cage and innovation in STEM and business are not too different from each other. Once ideas are generated, they must be evaluated. During the ideation process, technical feasibility and financial viability are not a consideration. Ideas should be generated with abandon. However, after the ideas are generated, they must be evaluated and the best ideas should be selected for formal conceptual testing.

2.3) Performance > Communication

Musicians, especially those that perform onstage, follow a set of performance principles that are also applicable to the communication of an idea. Communication of an idea, as well as the insight that informs an idea, and the ideation that creates the idea, are the three fundamental aspects of the innovation process in information technology innovation (Maddock 2011). Most of the principles of public speaking/public communication listed below are actually drawn from musical performance.

- Every gesture, movement, eye gaze, posture, stance, and any other physical movement (apart from speaking) needs to assist in the **communication** of the message and must **never distract** from the message. Feel an itch that needs to be scratched while speaking? Just don't, whether in public speaking or musical performance.
- Speak to the back of the room--aim for the top of the people's heads--with intermittent and occasional eye contact with individual audience members at points of emphasis. Whether public speaking or playing a horn, it's the same principle.
- Use silence for dramatic effect or getting attention or "owning the moment" at the beginning of the presentation, in the same way that a concert violinist will draw attention to the entrance of a gesture during a cadenza or before the commencement of the performance of a work.
- In some cases it is a good idea to memorize the presentation but don't sound like you are reading. Observe punctuation as an indication of phrasing, pitch inflection, and natural rest in between sentences. Vocalists are generally required to memorize their repertoire for performance, but they can never sound stilted.
- Keep all discussions high level but be prepared to drill down to the technical level at any moment (For example, have slides and explanations ready with technical explanations, and have slides and explanations ready with market research data). A musician should always be prepared to answer questions on the construction of a composition or the process of preparation of performance including interpretation choices. But those details should never take precedence over the larger message, nor should they be part of the performance.
- Bring your own projector, and prepare multiple backups of files saved in multiple formats. It's not over the top to bring a backup laptop and every adapter known to man. Computer musician performers know this best. One can never have enough backup technical solutions in case the primary system fails.
- Video record yourself giving a presentation in the same way that musicians tape their mock performances. It's hard to catch what needs to be improved while in the act of speaking or performing. A video recording, however, will painfully capture every flaw.
- When you have done the presentation 20-25 times, you are ready for your first presentation. It is always intriguing how musicians will practice countless hours to prepare for a performance of a ten-minute work, but entrepreneurs preparing a one-minute elevator pitch don't see the need to practice more than a few times. Whether a sonata or an elevator pitch, the performance of either must be practiced constantly.

2.4 Symphonic Thinking and Systems

"Symphonic thinking is a signature ability of composers and conductors, whose jobs involve corralling a diverse group of notes, instruments, and performers and producing a unified and pleasing sound. Entrepreneurs and inventors have long relied on this ability. But today Symphony is becoming an essential aptitude for a much wider swath of the population." (Pink 2006)

Basically, this is what computer music faculty do when they teach Max. In Max, musicians corral a large and diverse library of objects, integrate and coordinate them with operating systems platforms, utilities, hardware drivers, external hardware, and users, all in the pursuit of a musical goal. In short, we are developing a functional system in order to communicate an aesthetic message. When we inherit a patch and its system, we become systems analysts. In both roles, one has to imagine, put together, and see the individual pieces that cogently make up the larger whole (the larger system). In symphonic thinking, one sees relationships where others may not. In symphonic thinking, one sees the individual shapes and lines that make up a larger picture, where others may only see the larger picture. Conversely, symphonic thinkers can take small shapes and lines (objects) and put them together to represent something meaningful with resonation on the emotional level, regardless of the technical underpinning. They can integrate parts to create a solution. In symphonic thinking, one can recognize patterns in the system and make interdisciplinary, boundary crossing connections between disparate components. In many ways, symphonic thinking is a form of Wagnerian synthesis (*Gesamtkunstwerk*), which is the combination of all 19th-cetury Romantic styles in music with the visual arts and the dramatic arts folded into one unified piece of compelling total art. Wagner synthesized into one art form all of the above elements that preceded him and created *Gesamtkunstwerk* (total work of art). Wagner's *Gesamtkunstwerk* includes symphonic music, voice, mythology, poetry, visual art, stage scenery, and costumes.

One of the overriding competencies in information science is the ability to view and work with large systems and to be able to drill down and understand its inner workings, and how the details of those inner workings work with all the other details of inner workings, and ultimately how those details combine to create the larger system, product, or service that has emotional resonance, appeal, and relevance. This is necessary in the imagination stage, in the design and development of any system product or service, and it is also important in the analysis of a system, product, or service when it has to be refined, updated, and upgraded.

2.5) Symphonic Thinking > Organizational Management

The musical ensemble is a "Symphonic System." Inspired by Roger Nierenberg, this area of creative musical art uses the large musical ensemble as a metaphor for managing large organizations. In this metaphor, the conductor has a systems view, individual musicians with

their instruments are the smaller components that execute instructions, and the sheet music is the code. In short, the symphony orchestra's conductor role is a metaphor for executive leadership of a management information system.

In the application of the metaphor, these are the major analogies for demonstration of the key music concepts that can be directly applied to Management Information Systems: 1. leading by listening (enabling musicians or employees to synchronize by listening to each other vs. micromanaging from the podium), 2. nonverbal communication (inspiring, enabling, and facilitating from the stick or front office vs. giving verbal directives), 3. running legacy code, (interpreting symbolic music notation from the 19th century and porting the code to modern orchestral platforms), 4. the view (sound) from the silo (the specialist musician with their own vibrating system who executes one task really well vs. the view from the podium where all parts are heard and how they fit together and how that view/sound compares with the ideal version of a work), 5. a conductor's (or executive's) vision (for his/her symphonic system and getting the musicians to buy in to the vision, strategic goals, and organizational values).

3. CONCLUSION

This paper serves as a peek into the inner workings of music and arts curriculum within an information science and technology (STEM) curriculum. It is hoped that, more computer musicians can utilize their skill sets beyond the discipline of computer music/music technology and be afforded more opportunities in STEM education via the STEAM movement. It is hoped that computer musicians can proceed beyond disciplinary boundaries and recognize that there are no limits in the application of a computer musician's abilities to other technical disciplines in the field of higher education.

Music and fine arts faculty and curriculum within a STEM unit should not be viewed as unique, however. Although rare, there are other examples, such as composer Rand Steiger's 2010–2013 appointment to composer-in-residence at the California Institute for Telecommunications and Information Technology at UCSD, replacing composer Roger Reynolds. Lei Liang is currently the composer in residence through 2016. An example of a long-term music faculty appointment within a STEM unit is composer Tod Machover's tenure as a professor at the MIT Media Lab. A more advanced example of institutional integration of the arts faculty and curriculum into an information science curriculum can be seen at Indiana University Purdue University at Indianapolis (IUPUI), where the entire music department faculty, curriculum, and degree programs reside within the School of Engineering and Technology (IUPUI 2016).

4. REFERENCES

Bronson, Pro and Ashley Merryman (2010). *The Creativity Crisis*. [ONLINE] Available from http://www.newsweek.com/creativity-crisis-74665 [accessed February 7, 2016].

Calit2 (2010). *Technology Institute at UC San Diego Names Composer in Residence*. [ONLINE] Available from http://www.calit2.net/newsroom/release.php?id=1698 [accessed January 24, 2016].

Casner-Lotto, Jill and Linda Barrington (2006). *Are They Really Ready to Work? Employer's Perspectives on the Basic Knowledge and Applied Skills of New Entrants to the 21st Century U.S. work Force.* New York, NY: The Conference Board, Inc.

IUPUI (Indiana University-Purdue University Indianapolis), School of Engineering and Technology. [ONLINE] Available from http://www.engr.iupui.edu/departments/mat/ [Accessed January 24, 2016].

Kawasaki, Guy (2015). *Art of the Start 2.0*. Upper Saddle River, NJ: Pearson.

Kent, Corita (2008). *Learning By Heart: Teachings To Free The Creative Spirit*. New York, NY: Allworth Press.

Maddock, G., Uriarte, L., & Brown, P. (2011). *Brand New, Solving the Innovation Paradox*. Hoboken, NJ: John Wiley & Sons Ltd.

Michalko, M. (2006). *ThinkerToys* (Second Ed.). Berkley, CA: Ten Speed Press.

Nierenberg, Roger (2009). *Maestro*. New York, NY: Penguin.

Pink, Daniel (2006). *A Whole New Mind*. New York: Penguin Group.

Ries, Eric (2011). *The Lean Startup*. New York, NY: Crown Publishing Group.

President's Committee on the Arts and the Humanities, Reinvesting in Arts Education: Winning America's Future Through Creative Schools, Washington, DC, May 2011.

Wang, Ge (2014). "Principles of Visual Design for Computer Music." *Proceedings of the International Computer Music Conference 2014*. San Francisco, CA: International Computer Music Association.

The Things of Shapes: Waveform Generation using 3D Vertex Data

Kevin Schlei
University of Wisconsin-Milwaukee
3223 N. Downer Ave.
Milwaukee, WI 53211
kdschlei@uwm.edu

Rebecca Yoshikane
University of Wisconsin-Milwaukee
2400 E. Kenwood Blvd.
Milwaukee, WI 53211
yoshika5@uwm.edu

ABSTRACT

This paper discusses the implementation of a waveform generation system based on 3D model vertices. The system, built with the Metal API, reflects the GPU transformed vertex data back to the CPU to pass to the audio engine. Creative manipulation of 3D geometry and lighting changes the audio waveform in real time. The results are evaluated in a piece 'The Things of Shapes,' which uses unfiltered results to demonstrate the textural shifts of model manipulation.

1. INTRODUCTION

Visual-music systems have explored a variety of techniques to translate imagery into sound, in both pre-computational and post-computational contexts [1]. 3D modeling has contributed to this area, from rigid-body simulations to creative user interfaces. This paper outlines a method of connecting 3D model data to audio synthesis, initial results and evaluations, and further avenues for investigation.

The system presented in this paper is not a physical model simulation. Instead, 3D model vertices are treated as a creative stream of audio or control data. This allows for the exploration of odd geometries, impossible shapes, and glitches for imaginative results.

The generated audio is strongly linked to the visual product. By generating audio data directly from model vertices, the system creates an interaction mode where real-time object geometry manipulations alter the sonic result (see Figure 1). Changes to scene characteristics, like lighting, camera position, and object color, can contribute directly to the synthesized output. This allows for instant changes in timbre when switching between fragment shaders in real-time.

2. RELATED WORK

3D modeling is often used to create simulations of physical objects [2, 3, 4]. Models represent physical shapes, sizes, and material qualities.

3D models can also be representations of sound, or provide a UI for performance. *Sound Sculpting* altered synthesis parameters like chorus depth, FM index, and vibrato by manipulating properties of an object like position, orientation, and shape [5].

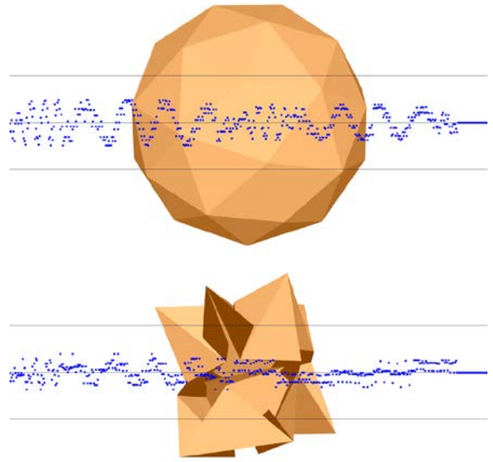


Figure 1. Vertex x and y coordinates create a waveform. The icosphere above is shown crushed below.

Non-geometric qualities of models, like textures and bump maps, can be used to simulate frictional contact, roughness, and impact events [6].

Auralization and spatialization techniques built around 3D model representations of physical space can use model data for physical simulation. *swonder3Dq* uses wave-field synthesis in conjunction with a virtually represented 3D space to model the radiation characteristics of sounding objects [7] .

Wave terrain synthesis is a method of interpolating over an arbitrary path that reads from a 2D array of amplitude values, often visualized in a 3D graph [8, 9, 10]. 'Wave voxel,' a 3D array lookup system offers a similar approach with an added axis [11]. These systems of interpolated values between vertex positions highlight how consideration of geometric shape can create variable sonic results.

The viability of GPU audio calculation has been evaluated in a number of studies. Gallo et al examined the cost of GPU vs. CPU operations on a number of audio tasks, including FFT, binaural rendering, and resampling [12]. GPU hardware limitations, including re-packing data into GPU recognized data formats, single input operations, and distribution for parallel computation, are addressed with 'Brook for GPUs,' a system designed for GPU streaming data calculations [13]. The outcomes are positive and show practical gains from assigning certain algorithms to

the GPU rather than the CPU.

3. VERTEX DATA MINING

A number of variables were identified as potentially valuable vertex data sources.

3D vertices' positions {x, y, z} are a primary data source, and they exist in multiple world spaces. The model contains its own model space, which is projected into a display world space, then flattened onto a viewport. The static model vertices were determined to be of little use, since the goal was to respond to display and user transformations. The projected and viewport coordinates proved to be dynamic and useful.

A vertices' normal vector reports which direction it is facing: whether it is pointed towards or away from a light source or eye (camera). In a typical lighting system, normal vectors determine how bright a face reflects the light source towards the eye.

In some render pipelines, a texture is stretched over the faces of a model. In others, the vertices have their own implicit color value. In both cases, the supplied lighting calculations alter the color values, which could be followed as a data stream.

Second order vertex properties, including velocity, color shift, brightness shift, etc., are under consideration for future implementations. Interpolation systems, such as those found in Haron et al [11], may also be explored.

Vertex Component	Value
Coordinate (projection space)	x, y, z
Coordinate (viewport)	x, y
Normal	x, y, z
Normal (angle to eye)	radians
Color	r, g, b
Color (value)	h, s, b

Table 1. Table captions should be placed below the table.

4. IMPLEMENTATION

4.1 Render Pipeline

The implementation aims to access 'cooked' vertex data: the transformed, projected, and fragment shaded result of a render pass of the GPU.

Like many graphic APIs, Metal requires two shaders to form a render pipeline: the vertex shader and the fragment shader. The vertex shader takes the model space vertices and transforms them into view projection space. This is also where user transformations (translate, rotate, scale) are applied.

The fragment shader (also known as pixel shader) is responsible for calculating the fragment (pixel) output after vertex rasterization. It interpolates between the vertex points to fill in the triangle faces seen on the display. Fragment shaders calculate lighting, texture sampling, and custom color functions. Unlike the vertex shader, which is called three times per triangle (once per vertex), the fragment shader may be called many thousands of times as it interpolates between vertices. This color data was mined to gain insight into which vertices are lit brightly, along with their color response to lighting conditions.

4.2 Accessing Vertex Data

Metal has two methods to retrieve data from the GPU: transform feedback and kernel functions.

Transform feedback is accomplished by passing an extra 'output' memory buffer to a vertex shader. The vertex shader writes the transformed vertices to the output buffer, which is then accessible to the CPU. However, the Metal API does not allow vertex shaders to pass data on for rasterization when they return output buffers. This means rasterization and transform feedback are mutually exclusive.

Kernel functions are parallel data computations that can be run on the GPU. They support writing to output buffers for CPU access. They also support multithreading for large data sets.

Unfortunately, neither solution leverages the existing render pipeline. This means two GPU pipelines are necessary: one to compute and reflect the vertex data back to the CPU, and a second that renders to the screen. While not ideal, it is not a doubling of computation time for the GPU, as explained in 4.3.

4.3 Compute Pipeline

A compute pipeline was created to calculate and retrieve the vertex data after transformation, projection, and lighting. It is passed identical copies of the vertex buffers that feed the render pipeline, as shown in Figure 2.

A kernel function was chosen to perform the calculation rather than a transform feedback vertex shader, due to its ability to split work into multiple threads. The kernel function body contains the same code as the render pipeline's vertex and fragment shaders, plus a few additional calculations for values like the viewport position.

Even though the same shader code is run in both pipelines, the GPU does not have twice the workload. A majority of the work of the render pipeline vertex shader is spent on implicit render actions, like depth testing and pre-fragment shader rasterization. Similarly, the fragment shader runs many more times than the number of vertices to calculate each pixel's color. The kernel function only performs a fraction of this total work, as seen in Table 2.

	Total	Avg.	Std. Dev.
Kernel	4.40 ms	72.10 μs	1.41 μs
Vertex	25.96 ms	432.64 μs	2.28 μs
Fragment	70.28 ms	1.17 ms	8.19 μs

Table 2. GPU shader computation times during 1000 ms of activity. Performed on a 2016 iPad Pro with A9X processor.

4.4 Pathway to Audio

The retrieved vertex data buffer is accessed after the kernel function has completed, at the end of each frame update. The frame ends by reading through the output buffer to pull the desired vertex component data. The data is written directly to a wavetable, which is continuously oscillated by the audio engine (libpd).

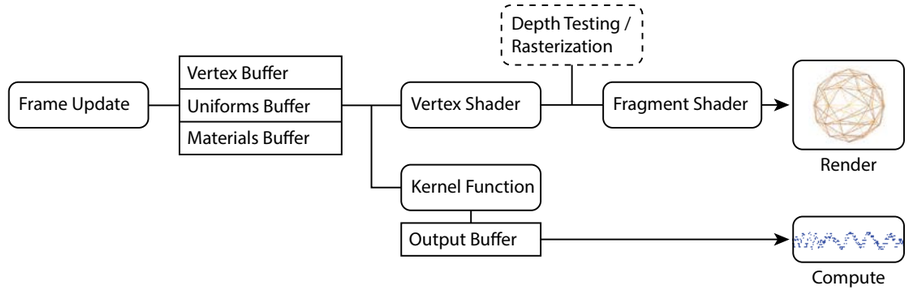


Figure 2. The render and compute pipelines for a frame update. Updated buffer data is passed to both shader pathways.

Typically a single vertex component is followed to generate a waveform. For example, the viewport (screen) x-position generates a waveform that changes as the model rotates, scales, or translates, and also when the camera position changes. Other components, such as color brightness and normal direction, created waveforms that reacted to other changes like lighting position or type.

Graphing combinations of components, like the normal x-value multiplied by the color brightness, can provide further variations on waveform responses that pull from more than one environmental change.

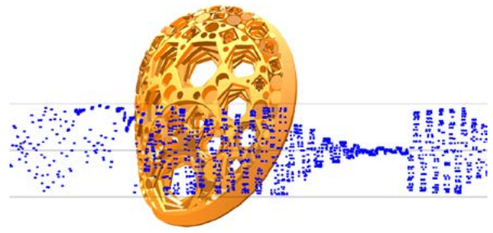


Figure 3. Normal values of vertices often produce areas of .

5. EVALUATION

5.1 Meshes App

A test application, 'Meshes,' was authored to test the performance and sonic results of the implementation. The user interface allows for 3-axis rotation of a model and translation away from the center point. A simple momentum system lets the 3D model be 'thrown' around the world space. A slider sets the wavetable oscillation frequency.

5.2 General Observations

The ability to generate audio data from 3D model data shows strong promise in a few areas. First, the offloading of complex parallel data calculation to the graphics card frees the CPU to perform more audio functions. This is especially beneficial on mobile devices.

Second, the direct link from model shapes and lighting systems presents a novel interaction mode for sound generation. The variety of sonic outputs from different 3D model shapes and lighting formulas allows a 3D modeling artist to creatively engage with the sound generation.

5.3 Vertex Component Variations

The different 'cooked' vertex components resulted in a variety of wavetable results. Projected coordinate values (x, y, and z position) created subtle harmonic shifts when rotated about their axes. Translating the 3D model away from the camera position, however, produced no change. This is because the projected model data remains steady in its world-space position. Meanwhile the viewport position (screen x and y) would shrink and expand as the model moves closer and farther from the camera.

The normal values often created areas of visible waveform continuity. For example, a prosthetic cap model (Figure 3) produced cyclical patterns where its geometric cutouts produced a cylindrical shape. Oscillated slowly, these patterns produced shifting rhythmic cycles.

The color brightness value shows great potential as a way to drastically alter the sonic output of a model. By switching between fragment shaders, shown in Figure 4, the brightness values can be shifted, sloped, and quantized.

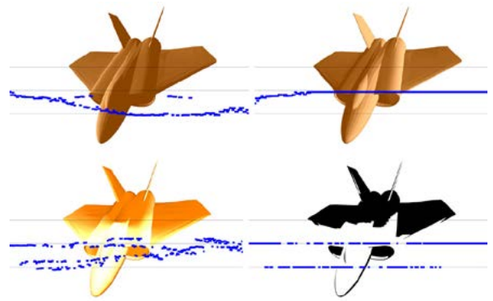


Figure 4. The lighting direction or fragment function changes vertices' brightness values, graphed here using the 'realship' model.

5.4 Duplicate Pipeline Performance

The splitting of vertex calculations into two pipelines (render and compute) has some drawbacks, but also some major positives.

One drawback is the lack of access to depth testing that

occurs during the render pipeline (see Figure 2). Faces of objects that fail depth tests, i.e., are 'behind' other faces, are not fragment shaded or drawn to screen. Since that process is automatically performed between the vertex shader and fragment shader in a render pipeline, it is not available to the compute kernel function. These changes to the vertex data may have allowed for interesting effect possibilities.

A significant benefit of splitting work between the render and compute pipeline is the possibility of decoupling audio updates from screen updates. Performance tests, like Table 2, indicate that the GPU spends around 95.5% of a frame update on the render pipeline, vs. just 4.5% on the compute pipeline. This shows that the compute pipeline could be run separately at a much faster rate, perhaps called directly from the audio buffer callback.

5.5 CSIRAC and the 'blurt'

CSIRAC was the first computer to generate digital audio by sending its bitstream directly to an amplifier [14]. The direct mapping of vertex data to a waveform outlined in this research is similar to CSIRAC's sonification of computer data.

An interesting historic note is the 'blurt:' a short, recognizable loop of raw pulses which CSIRAC programmers added to the end of a program. Lacking a display terminal, this aural cue helped signify when a program had finished.

The sonification of vertex data also acted as a helpful debugging tool. For example, when listening to the projected x-coordinate of a model translated entirely off-screen, one might expect to hear silence. Surprisingly, the waveform persisted. This lead to the realization that the flat, viewport coordinates and the model's projection-space coordinates were separate data.

Furthermore, viewing the generated waveform illustrated how unorganized model vertices could be. Simple geometric shapes, created in commercial 3D modeling software, were shown to have no discernible pattern of face order, as shown in Figure 5. This is not an issue in the implementation, but rather highlights the naturally occurring structures of 3D models.

Figure 5. Four iterations of drawing an icosphere, where consecutive vertices are allowed to be drawn.

5.6 The Things of Shapes

'The Things of Shapes' is a piece that uses the unfiltered output from the vertex wavetable to create a collage composition with a frenetic character. The gestures are articulated by both automated and user-driven manipulation of 3D models. The 3D models used include simple geometric shapes (cube and icosphere) and complex models (realship). The majority of the piece used only the x-coordinate property of vertices for waveform generation.

One automated manipulation was a noise function which randomly fluttered vertex positions, with a variable spread, as seen in Figure 6. The result added noise to the waveform, but even at high levels of flutter a discernible waveform timbre could be maintained and controlled.

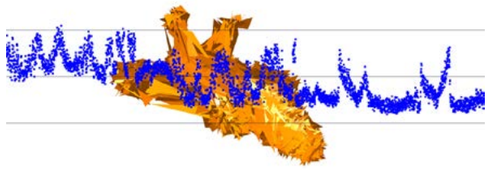


Figure 6. Vertices are shifted by a random amount each frame to add noise to the shape and waveform.

Next a modulo function was used to crush vertices towards the zero-point of the coordinate space, as seen in Figure 1. This function was cycled to ramp between unaltered model shapes and crushed shapes. This caused some models to pulse from zero to their original scale.

User-driven model manipulation was achieved through touch screen interactions and physics simulations. Rotation, offset, and scale of the models were attached to touch panning gestures. These in turn were given momentum and resistance properties to allow for natural deceleration of position and rotation. This formed a major influence on the gestural quality of the final piece. Slowly rotated or shifted shapes produced steadily changing timbres in the waveform. 'Thrown' shapes, where the model rotated at some distance around the center of the projected space like a tetherball, brought the vertices into and out of the viewport. This cyclical appearing and disappearing produced a fluttering sound that decelerates towards a steady tone.

5.7 Improvements and Future Work

Synchronization of audio buffer callbacks and kernel function calculations is the first priority of future implementations. In addition to being lower latency than the current display update rate of 60Hz, synchronization could allow for audio-rate data to be fed into the kernel function. This would allow for smooth audio calculation from within the kernel function, rather than the control-rate updates currently implemented. Another option would be to pursue streaming implementations such as those found in Brooks et al [13].

Systems of generating audio that do not rely on direct mapping could create new synthesis possibilities. Instead of using wavetable synthesis, an internal oscillation method could be devised and continuously output. This may be based on interpolating between weighted vertices, or using relational analysis of the entire collection of vertices to drive synthesis parameters.

Model manipulation could be improved with a variety of methods to alter model geometry and fragment calculation. Advanced object deformation, such as fabric or viscosity mesh simulations, could be sonified. More graphic pipeline functions, including masking, blending, bump mapping, etc., could be implemented as creative methods of generating data.

Geometry shaders are relatively new shaders where the GPU can generate new vertices from the originally provided vertices. The Metal API currently does not support geometry shaders. Two-stage vertex calculation has been offered as a workaround for this.

6. CONCLUSIONS

A method for accessing projected and fragment shaded vertex data from a GPU has been outlined. Initial observations show a successful split between rendering and compute pipelines, with the possibility of further decoupling to improve audio calculation latency. A prototype application demonstrated how sonic changes follow the transformation of models fed through the system. 'The Things of Shapes' takes that tool and assembled a collage of shape-driven sounds and phrases.

Acknowledgments

The authors would like to thank the Office for Undergraduate Research at the University of Wisconsin-Milwaukee for their support and funding for this project.

7. REFERENCES

[1] G. Levin, "Painterly interfaces for audiovisual performance," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[2] J. F. O'Brien, C. Shen, and C. M. Gatchalian, "Synthesizing sounds from rigid-body simulations," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2002, pp. 175–181.

[3] N. Raghuvanshi and M. C. Lin, "Interactive sound synthesis for large scale environments," in *Proceedings of the 2006 symposium on Interactive 3D graphics and games*. ACM, 2006, pp. 101–108.

[4] K. Van Den Doel, P. G. Kry, and D. K. Pai, "FoleyAutomatic: physically-based sound effects for interactive simulation and animation," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 537–544.

[5] A. Mulder and S. Fels, "Sound sculpting: Manipulating sound through virtual sculpting," in *Proc. of the 1998 Western Computer Graphics Symposium*, 1998, pp. 15–23.

[6] Z. Ren, H. Yeh, and M. C. Lin, "Synthesizing contact sounds between textured models," in *Virtual Reality Conference (VR), 2010 IEEE*. IEEE, 2010, pp. 139–146.

[7] M. Baalman, "swonder3Dq: Auralisation of 3D objects with wave field synthesis," in *LAC2006 Proceedings*, 2006, p. 33.

[8] Y. Mitsuhashi, "Audio signal synthesis by functions of two variables," *Journal of the Audio Engineering Society*, vol. 30, no. 10, pp. 701–706, 1982.

[9] A. Borgonovo and G. Haus, "Sound synthesis by means of two-variable functions: experimental criteria and results," *Computer Music Journal*, vol. 10, no. 3, pp. 57–71, 1986.

[10] R. C. Boulanger, *The Csound book: perspectives in software synthesis, sound design, signal processing, and programming*. MIT press, 2000.

[11] A. Haron and G. Legrady, "Wave voxel: A multimodal volumetric representation of three dimensional lookup tables for sound synthesis."

[12] E. Gallo and N. Tsingos, "Efficient 3d audio processing on the gpu," in *ACM Workshop on General Purpose Computing on Graphics Processors*, 2004.

[13] I. Buck, T. Foley, D. Horn, J. Sugerman, K. Fatahalian, M. Houston, and P. Hanrahan, "Brook for GPUs: stream computing on graphics hardware," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 777–786.

[14] P. Doornbusch, "Computer sound synthesis in 1951: the music of CSIRAC," *Computer Music Journal*, vol. 28, no. 1, pp. 10–25, 2004.

Sound vest for dance performance

Felipe Otondo
Universidad Austral de Chile
felipe.otondo@uach.cl

Rodrigo Torres
Universidad Austral de Chile
rodrigo.torres@uach.cl

ABSTRACT

The importance of spatial design in music has become more noticeable in recent years mostly due to the affordability of improved and powerful software and hardware tools. While spatial audio tools are extensively used nowadays in different kinds of musical applications, there are very few examples of mobile sound systems especially conceived for the performing arts. An original sound vest prototype featuring original costume design, a hybrid full-range loudspeaker array and an improved acoustic response was designed and implemented using data gathered from anechoic measurements and interviews with performers and audiences. Future developments of the system will consider the implementation of an extended multi-channel platform that will allow the possibility of exploring sonic and spatial relationships generated by several mobile sound sources on stage in connection with a multi-loudspeaker diffusion system.

1. INTRODUCTION

The development of cheaper and powerful software and hardware tools has allowed the topic of spatialised sound in music to gain a considerable momentum in recent decades [1, 2, 3]. The use of multi-channel sound systems for films, site-specific installations, videogames has increased awareness among audiences and artists about the creative possibilities of spatialised sound [3, 4, 5]. In recent decades the use of spatial audio tools has expanded to the performing arts, whereby performers, composers and technology developers have started to integrate mobile sound devices as organic components of music and dance projects [6, 7, 8]. While there has been some innovative dance projects involving mobiles sound systems, there is still a lack of flexible software and hardware tools that will allow artists to effectively relate creative features of music composition and dance choreography in collaborative projects. In this paper, the design, implementation and optimization of an original body-worn sound system is discussed, taking as a point of departure an interdisciplinary research approach which involved choreographers, performers, technology developers and musicians.

2. WIRELESS BODY-WORN SOUND SYSTEM

2.1 Design and implementation

Different kinds of artists and technology developers have carried out various kinds of projects involving the design and implementation of mobile sound systems [9, 10, 11]. Hahn and Bahn designed and implemented an original interactive platform for dance that included a 'sensor-speaker performer' interface, which located and reproduced sounds directly from the performer's body using two independent audio channels to feed the system [12]. From the documentation available about the system it is evident that the large size and shape of the system´s interfaces constrained considerably the movements and flexibility of performers on stage [13]. In recent years Johannes Birringer and Michèle Danjoux at DAP-Lab at Bruynel University in England have also designed and implemented different types of wearable mobile sound systems for various performance projects [14]. Aiming to enhance relationships between physical and virtual spaces, they designed and implemented original sound costumes and portable sound props to be used by performers as part of different kinds of multi-media productions. Possibly their most ambitious work involving mobile sound devices was the piece *UKIYO*, premiered in November 2010 at the Sadler's Wells' Lilian Baylis studio in London [15]. The piece was conceived as a site-specific multimedia installation where 'dancers and musicians perform simultaneously with digital objects that mutate; garments are custom-built for sound in motion' [8]. During the performance of the piece a singer and a dancer worn sound vests especially designed for the project while an actor carried two portable loudspeakers on a yoke. In the opinion of the author, the mobile sound systems used for the piece revealed during the performance technical and practical problems that constrained considerably the artistic potential and flexibility of the work. The first noticeable issue identified was the fact that the body-worn systems used by performers were large and had to be connected to a power supply, posing obvious limitations for actors, singers and members of the audience in the performing area. A second problem identified during the show was the limited acoustic power of the sound devices worn by performers. During the performance the projected sound by mobile sound sources was frequently masked by the voices of actors and sounds radiated by the PA system in the room. Taking into account some of the acoustic and practical limitations of mobile sound systems described above, it was decided to design an original wireless body-worn prototype to be tested and implemented with dancers in-situ [16]. The main objective of the project was to develop a robust and acoustically reliable system that could be adjusted to the requirements of performers in different kinds of artistic situations. The designed system had to be capable of effectively radiating sound in small and medium-size performance venues and flexible enough to allow dancers to carry out conventional movements in both standing positions and on the floor. The system considered two loudspeaker units located in the front and back of the performer's torso, a two-channel *Maxim* 25 Watt amplifier fed by 12 Volt batteries and a two channel 2.4 GHz Bluetooth transmitter with receiver set. One of the main challenges of the prototype design was the construction of small loudspeaker cabinets that will not impede dance movements and at the same time provide enough sound power to effectively radiate sound across a medium size venue.

2.2 System adjustments

Different types of tests to assess the flexibility and robustness of the body-worn system were carried out with dancers in a studio. After various trials performers were overall satisfied with the design of the system, but had certain concerns regarding the position of the back torso loudspeaker. One dancer noted that this loudspeaker restricted considerably the range of body movements, especially for actions taking place on the floor. In order to increase the control over radiated sounds by the performer, the dancer suggested to include loudspeakers attached to the arms of the performers. These suggestions were taken on board and it was therefore decided to modify the original architecture of the prototype by removing the rear speaker to include a pair of small speakers on both forearms of the dancer. As a way of finding the most suitable pair of loudspeakers for the performer's forearms, several kinds of 2-inch full-range loudspeakers units were tested and measured in an anechoic chamber. Frequency response and sensitivity measurements showed that the loudspeaker unit with the best overall acoustic performance was the *Vifa NE65W* [17, 18]. The next step in the optimization process was to find suitable cabinets for the chosen loudspeaker unit, focusing on two main design criteria. The first criterion was to maximize the acoustic power and frequency response of the *Vifa NE65W* units for small and medium size dance studios. The second criterion was to make the size of the cabinets as small as possible in order to allow the performer carry out regular dance movements in standing positions and on the floor. Anechoic measurements of the *Vifa* NE65W loudspeaker mounted on different size cabinets showed that for volumes below 250 cm^3 the variations in the frequency response and sensitivity of the loudspeakers were minor. In order to optimize the size of the forearms loudspeakers it was therefore decided to build the smaller cabinet size that would fit the *Vifa* NE65W speaker units. The volume of this cabinet was 100 cm^3 and the measured sensitivity of the loudspeaker system with this cabinet unit was 79.9 dB (1m/1W), within the range of the sensitivity of small conventional home studio loudspeaker systems and within the original expectations for the system. In order to improve the overall performance of the wearable application other aspects of the system were also modified. Regular commercial rechargeable battery units of the prototype were replaced with lithium-ion batteries, which extended the functioning duration of the system in 30 minutes and were significantly lighter than conventional commercial rechargeable batteries. Another improvement of the optimized system was the increased power of the built-in electric amplifier. A new more powerful amplifier with 30 Watt RMS per channel was added. This especially designed amplifier could easily drive two more extra loudspeakers, allowing the possibility of expanding the capacity of the current system in the future. Figure 1 and 2 show a dancer wearing the optimized body-worn system during tests in a dance studio in Valdivia, Chile.

Figure 1. Frontal view of the body-worn loudspeaker system during a dance demonstration.

Figure 2. Rear view of the body-worn loudspeaker system during a dance demonstration.

3. TESTS WITH PERFORMERS

An early demonstration of the system took place at the IX Ibero-American Congress on Acoustics in Valdivia, Chile. During the event a short dance improvisation was performed by a dancer wearing the system in a 200 m^3 dance studio. During the presentation the dancer exemplify numerous kinds of movements while the system played a two-channel mix created using different types of sounds materials. After the demonstration several members of the audience were asked about their impressions regarding the acoustic performance of the wearable sound system. Responses showed that the expressive character of the application, as well as the striking effect of the embodiment of movement and sound on and off stage, impressed most respondents. Quizzed about the acoustic power of the system, most participants considered that the application was easily capable of covering the size of a small and medium size dance studio. Questioned about the quality of the vest's reproduced sound, most respondents were positive about the overall functioning of the system, but noted that, the quality of the reproduced seemed to be very dependent on the type of sound material played [8, 15]. Another interesting aspect of the system mentioned by several respondents was that when the dancer performed in close proximity to the audience, the body-worn system was capable of creating a very intimate and subtle acoustic experience. The dancer was also questioned about his experience using the sound vest system. A considerable improvement in terms of flexibility and weight was noted in comparison to the original prototype, mostly obvious in regular movements in standing and floor positions. The performer also noted that, when in close proximity to the public, communication with the audience seemed to be enhanced by the use of the body-worn system and the possibility of being able to radiate sounds through his arms. As in similar dance projects where performers had control over sounds on and off stage, in this case the performer felt that he could play a more important role in the implementation of the piece by conceiving his artistic role as blend between a dancer and a musician [7, 18, 19].

A second demonstration of the system took place during a residency with dancers and choreographers that took place in the city of Valdivia. The demonstration was done by a dancer using the system playing synthesized tones in a dance studio. A discussion with dancers and a choreographer took place after the demonstration where various aspects of the application were examined. Initially it was agreed by most participants that the system provided a subtle sonic component to the dance, which was very dependent on the kinds of sound materials used to feed the system. It was also noted by the choreographer that it was evident the type of sound materials reproduced has a direct impact on the performer's response to the dance. Considering a new situation where the performer is no longer only a dancer, but also a musician projecting sounds through his/her torso and arms, it was clear that there has to be a process of reflection by the choreographer, performer and composer involved in the project in order to understand the new role of the dancer in the piece. When demonstrating the system it also became clear that single raw sound materials worked much better than textures of sounds that had been carefully crafted forehand. The complex shape and architecture of the system on the body of the performer and the important influence of the movement on the perceived sounds requires clean and transparent sounds that can be easily shaped during performance.

4. DISCUSSION

The aim of this study was to design and test a robust and acoustically reliable mobile sound system that could be easily adapted to the requirements of dancers in different types of performance environments. The main challenge of the project was to balance artistic, technical and practical specifications of a body-worn sound system suitable for contemporary dance practice. Early tests and demonstrations showed that sound wearable devices are very effective tools to establish close links with audiences during performances [20, 21]. Further studies with the designed system will explore ways of assessing this impact in different kinds of performance scenarios using suitable evaluation methods like listening tests with trained panels [22], context-methods surveys [3] or perceptual studies exploring spatial features music performance in concert halls [23, 24]. The impact that the sound vest has on the way performers conceive their role in a dance or music piece is also an important aspect to be investigated in future research activities. Evidence shows that performers that participated in projects involving the use of mobile sound devices consider that the use of these systems has a positive impact in a piece's artistic process, enhancing creative relationship between the choreographer, performers and the composer [7, 8]. Following developments of a related research project with students from various artistic backgrounds, further studies will explore different kinds approaches for successfully integrating compositional and choreographic strategies by relating specific body movements with sonic spatial attributes in a dance piece [20, 25]. By integrating corporeal and sonic movement, the body-worn sound system allows the composer, choreographer and dancer to investigate aesthetic relationships that go beyond the traditional associations found in dance and music performance. An interesting challenge for future performance with the system implemented in this study will be to develop a suitable framework where particular spatial and timbral features of multi-channel electroacoustic music performance can be successfully translated into a dance performance environment. Some of the early pilot tests mentioned above showed that this is an intricate issue because sound materials reproduced through stationary speakers are perceived by listeners in a very different way when they are projected through sound sources attached to a moving body. The performer's body drastically shapes the input to the sound system, making the acoustic output a complex modulated sound shaped by dance movements of the performer and the position of the loudspeaker units in the body of the performer. This implies that in order to make mobile systems work effectively in different kinds of performance environments; it is important to try out sound materials in a realistic performance environments rather than in acoustically treated spaces. In this context it is important to understand that when using the system in most dance performance situations, raw sounds, with little or no timbral and spatial processing, will work better in mobile sound systems than carefully composed sound materials, which is normally obscured by spatial and timbral modulations derived from the performer's movements.

The use of several performers wearing sound vests on stage linked with a multi-channel loudspeaker sound reproduction platform could be a natural development of this project. Early tests with two pairs of commercial wireless loudspeakers carried by actors synchronized with a four-channel fixed system revealed the potential of mobile sound sources to effectively enhance various performance features of multi-channel electroacoustic music that are normally lost in most concert situations [1, 20]. Trials with 8-channel hybrid systems as the ones mentioned above showed that by means of blending and contrasting multiple real and virtual sound sources on stage a greater sense of intimacy for audiences can be achieved, as well as an effective spatial counterpoint between travelling sounds sources on stage and projected sounds through a fixed sound reinforcement system. The artistic, perceptual and practical implications of such hybrid arrangements will be studied in future developments of the project presented here.

5. REFERENCES

[1] F. Otondo, "Contemporary trends in the use of space in electroacoustic music," *Organised Sound,* vol. 13, no. 1, pp. 77-81, 2008.

[2] N. Peters, G. Marentakis and S. McAdams, "Current Technologies and Compositional Practices for Spatialization," *Computer Music Journal,* vol. 35, no. 44, pp. 10-27, 2011.

[3] S. Wilson and J. Harrison, "Rethinking the BEAST: recent developments in multichannel composition at Birmingham ElectroAcoustic Sound Theatre," *Organised Sound,* vol. 15, no. 3, pp. 239-250, 2010.

[4] F. Otondo, "Creating Sonic Spaces: An Interview with Natasha Barrett," *Computer Music Journal,* vol. 31, no. 2, pp. 10-19, 2007.

[5] E. Stefani and K. Lauke, "Music, space and theatre: site-specific approaches to multichannel spatialisation," *Organised Sound,* vol. 15, no. 3, pp. 251-259, 2010.

[6] C. Wilkins and O. Ben-Tal, "The embodiment of music/sound within and intermedia performance space," *Proceedings of the 5th International Conference on Digital Arts*, London, pp. 19-24, 2010.

[7] A. Stahl and P. Clemens, "Auditory Masquing: wearable sound systems for diegetic character voices," *Proceedings of the Conference on New Interfaces for Musical Expression,* pp. 427-430, 2010.

[8] J. Birringer and M. Danjoux, "Audible and inaudible choreography," *Etum - E-journal for Theatre and Media,* vol. 1, no. 1, pp. 9-32, 2014.

[9] A. Tanaka and P. Gemeinboeck, "A Framework for Spatial Interaction in Locative Media," *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression*, Paris, 2006.

[10] G. Schiemer, and M. Havryliv, "Pocket Gamelan: Swinging Phones and ad hoc Standards," *Proceedings of the 4th International Mobile Music Workshop,* Amsterdam, 2007.

[11] Steffi Weismann's website. www.steffiweismann.de, 2016.

[12] T. Hahn and C. Bahn, 2002, "Pikapika - The Collaborative Composition of an Interactive Sonic Character," *Organised Sound,* vol. 7, no. 3, pp. 229-38, 2002.

[13] Curtis Bahn and Tomie Hahn. Website. www.arts.rpi.edu/bahnc2/Activities/SSpeaPer/SSpeaPer.htm, 2016.

[14] J. Birringer, "Moveable Worlds/Digital Scenographies," *International Journal of Performance and Digital Media,* vol. 6, no. 1, pp. 89-107, 2010.

[15] *Ukyo website*, http://people.brunel.ac.uk/dap/Ukiyo_Sadlerswells.html, 2016.

[16] Greenlight AV, "Research and development of body worn speaker systems for Lancaster University," Lancaster University internal report, 2010.

[17] *Tymphany.* www.tymphany.com/transducers/transducer-search-results/?keywords=ne65w

[18] F. Otondo, "Wireless Body-worn Sound System for Dance and Music Performance," *Organised Sound,* vol. 20, no. 3, pp. 340–348, 2015.

[19] S. Lanzalone, "Hidden Grids: Paths of Expressive Gesture between Instruments, Music and Dance," *Organised Sound, vol.* 5, no. 1, pp. 17-26, 2000.

[20] F. Otondo, "Mobile sources in two music theatre works," *Proceedings of the International Computer Conference*, New York, pp. 446-449, 2010.

[21] J. Birringer and M. Danjoux, "The Sound of Movement Wearables: Performing UKIYO," *Leonardo*, vol. 46, no. 3, pp. 232-40, 2013.

[22] S. Bech and N. Zacharov, *Perceptual Audio: Evaluation theory*, *Method and Application*. John Wiley, 2006.

[23] H. Lynch and R. Sazdov, "An Investigation into Compositional Techniques Utilized for the Three-dimensional Spatialization of Electroacoustic Music," *Proceedings of the Electroacoustic Music Studies Conference*, New York, 2011.

[24] R. Sazdov, R., "The Influence of Subwoofer Frequencies Within a Multi-channel Loudspeaker Configuration on the Perception of Spatial Attributes in a Concert-hall Environment," *Proceedings of the International Computer Music Conference*, Huddersfield, UK, 2011.

[25] F. Otondo, "Using spatial sound as an interdisciplinary teaching tool," *Journal of Music, Technology and Education,* vol. 6, no. 2, pp. 179-190, 2013.

WebHexIso: A Customizable Web-based Hexagonal Isomorphic Musical Keyboard Interface

Hanlin Hu
University of Regina
hu263@cs.uregina.ca

David Gerhard
University of Regina
gerhard@cs.uregina.ca

ABSTRACT

The research of musical isomorphism has been around for hundreds of years. Based on the concept of musical isomorphism, designers have created many isomorphic keyboard-based instruments. However, there are two major concerns: first, most instruments afford only a single pattern per interface. Second, because note actuators on isomorphic instruments tend to be small, the hand of the player can block the eye-sight when performing. To overcome these two limitations and to fill up the vacancy of web-based isomorphic interface, in this paper, a novel customizable hexagonal isomorphic musical keyboard interface is introduced. The creation of this interface allows isomorphic layouts to be explored without the need to download software or purchase a controller. additionally, MIDI devices may be connected to the web keyboard to display the isomorphic mapping of note being played on a MIDI device, or to produce control signals for a MIDI synthesizer.

1. INTRODUCTION

Since Euler introduced *Tonnetz* in 1739 [1], mathematician, composers, computer scientists and instrument designers are interested in musical isomorphism, which presents algorithms of arrangement of musical notes in 2-dimensional space so that musical constructs (such as intervals, chords and melodies, etc.) can be played with same fingering shape regardless the beginning note [2].

Based on this concept of musical isomorphism, there have been a number of keyboard-based interfaces (instruments) being built over the last hundred years. At the beginning, the keys on the keyboard were designed in the shape of a rectangle resembling that of the traditional keyboard. This is called "square" or "rectilinear" isomorphism. Since the limitations of the square isomorphism (*e.g.* the degenerating of layouts, which is passing by some notes in a equal temperament with particular layouts [2]), interface designers prefer hexagonal shaped keys on the keyboard over the last decade. They can be categorized to hardware and software. As for hardware, there are the AXiS keyboards, Opal, Manta, Tummer and Rainboard [3]; As for software, there are Hex Play (PC) [4], Musix Pro (iOS) [3], Hex OSC Full (iOS) [5] and Hexiano (Android) [6] .

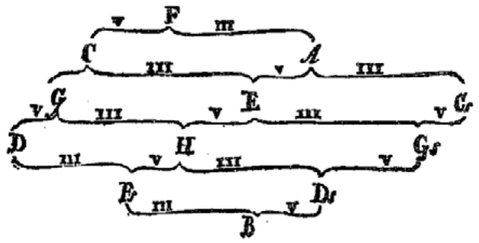


Figure 1: Euler's Tonnetz

In addition to the limitation of square isomorphism, there are two main constraints of isomorphic keyboard design: firstly, most of the interfaces can provide only one particular isomorphic layout, which means the layout is not changeable or in the other word, the device is not customizable, therefore the performers or composers are "locked in" to a single layout. Although learning a single layout may be desirable in some circumstances, one of the main advantages of isomorphic keyboard design is the fact that many different layouts with different harmonic relationships are available in the same framework. For this reason, we think that the limitation of a single isomorphic layout on most hardware instruments is a significant disadvantage which should be addressed.

Secondly, when performers or composers play on the physical appearance of an isomorphic keyboard, their hands can easily block the display so that the name or colour of keys is not easy to see. For traditional single-layout instruments, this is not an issue because performers memorize and internalize key-actuator positioning, but for reconfigurable digital instruments, where colour may be the only indication of the function or note of an actuator, the problem of actuator occlusion may be considered significant. One possible solution would be to separate the display of the reconfigurable keyboard from the actuators. Although this separation may be considered a step back in terms of usability and control/display integration, it may serve to facilitate exploration of this new class of reconfigurable interfaces. Further, although it is possible to plug a MIDI device into an iPad and play isomorphic software such as Musix Pro, practically speaking, because many MIDI devices draw power over the USB port, it is sometimes impractical or impossible to connect a MIDI device to an iPad or other external display.

In this paper, we present a novel web-based hexagonal isomorphic musical keyboard interface, which is customizable, scalable, and MIDI enabled. It can be used as a software instrument, a composition device, an educational tool of musical isomorphism, and an assistive screen interface for performance.

2. MUSICAL ISOMORPHISMS

The word "isomorphism" has the prefix "iso," which means "equal," and an affix term "morph," which means "shape." Isomorphism, then, refers to the property of having an identical shape or form. The concept of isomorphism applied to music notations is that for an isomorphic arrangement of notes, any musical construct (such as an interval, chord, or melody) has the same shape regardless of the root pitch of the construct. The pattern of constructs should be consistent in the relationship of its representation, both in position and tuning. Corresponding to transposition invariance, *tuning invariance* [1] is another requirement of musical isomorphism. Most modern musical instruments (like the piano and guitar) are *not* isomorphic. The guitar in standard tuning uses Perfect Fourth intervals between strings, except for the B string which is a Major Third from the G string below it. Because of this different interval for one pair of strings, the guitar is not isomorphic.

Isomorphic instruments are musical hardware which can play the same musical patterns regardless of the starting pitch. Isomorphic arrangements of musical notes introduce a number of benefits to performers [7]. The most notable of these is that fingerings are identical in all musical keys, making learning and performing easier. Modern instruments which display isomorphism include stringed instruments such as violin, viola, cello, and string bass [3]. It should be noted in this case that although the relative position of intervals is the same for every note on the fingerboard of a violin, the relative size of each note zone may change, with the notes being smaller as you move closer to the bridge of the instrument. The traditional piano keyboard is not isomorphic since it includes seven major notes and five minor notes as a 7-5 pattern.

Thanks to this 7-5 pattern design, performers can easily distinguish in-scale and out-of-scale notes by binary colours, but the performer must remember which white notes and which black notes are in the scale in which they are performing. Because the piano is not isomorphic, different fingerings and patterns are required when performers play intervals and chords in different keys. This is one of the reasons that the piano is difficult to learn: each musical construct (*e.g.* the Major scale) must be learned separately for each key (*e.g.* C Major, G Major, F Major *etc.*)

The first physical appearance of an isomorphic layout was decided by Hungarian pianist Paul von Jankó in 1882 [8]. The *Jankó keyboard* shown in Fig. 2 was originally designed for pianists who have small hands that can cause fingering difficulties when stretching to reach the ninth interval, or even the octave, on a traditional keyboard. By setting every second key into the upper row and shaping all keys identically, the size of the keyboard in the horizontal direction shrinks by about half within one octave. After making three duplicates, the performer can play intervals or chords by putting the fingers up or down to reach the desired notes. Each vertical column of keys to the adjacent column are a semitone away, and the horizontal row of keys to the adjacent row is a whole step away. This design never became popular since performers are not convinced of the benefits of this keyboard and they would instead have to spend more time learning a new system [9].

[1] Tuning invariance: where all constructs must have identical geometric shape of the continuum

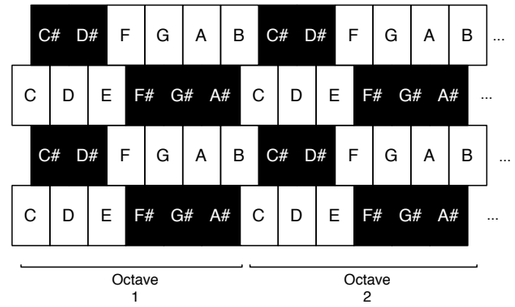


Figure 2: Jankó Keyboard tessellation

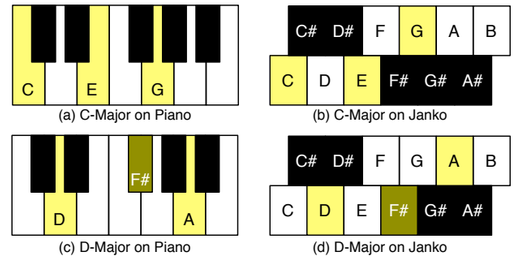


Figure 3: Isomorphism in the Jankó keyboard as compared to polymorphism in the piano keyboard.

This arrangement of notes on the Jankó keyboard is isomorphic because a musical construct has the same shape regardless of key. Consider a Major triad (Fig. 3). the C-Major triad has the notes C-E-G, while the D-Major triad has the notes D-F♯-A. On the piano keyboard, these triads have different shapes, but on the Janko keyboard (and on any isomorphic keyboard) these triads have the same shape. In fact, every major triad has the same shape on an isomorphic keyboard.

There are three reasons why hexagonal shaped keys are better than rectangular shaped keys:

1. The rectangular shaped keys of an isomorphic keyboard does not meet the requirement of transposition invariance perfectly, because the Euclidean distance between two keys is not identical. For example, in Fig. 2, the distance between C and D in horizontal direction does not equal the distance between C♯ and D in vertical direction. However, the regular hexagonal shaped keys make the distances identical.

2. It is easy to find three adjacent keys to close into a triangle on Jankó layout, such as C-D-C♯. This relationship has been modelled with equilateral triangle by Riemann as triangular Tonnetz in Fig. 4, which is explored from Tonnetz [11]. Since the regular hexagon is the dual of the equilateral triangle, for each key, the hexagonal shape is good to present this relationship.

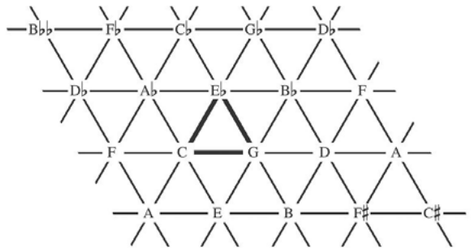


Figure 4: Tonnetz as regularized and extended by Riemann and others [10]

3. Each hexagon has six adjacent hexagons, while each square only has four adjacent squares. More adjacent notes means more harmonic connects and a more compact note arrangement for the same number of notes.

3. WEBHEXISO

WebHexIso is a novel customizable web-based musical isomorphic interface. There are few existing web-based musical isomorphic interface available online such as "CanvasKeyboard" [12]. However, the "CanvasBoard" is uncustomizable interface with a particular layout — Wicki-Hayden [13].

3.1 Basic Design

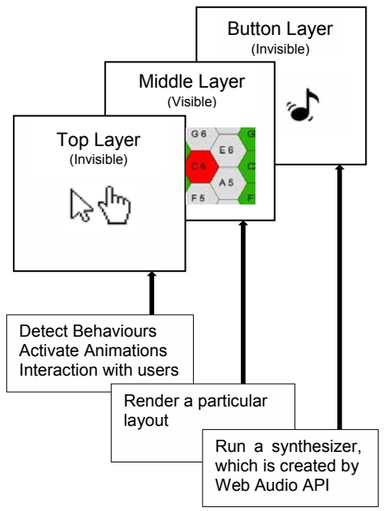
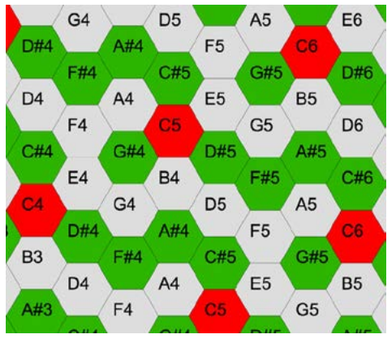


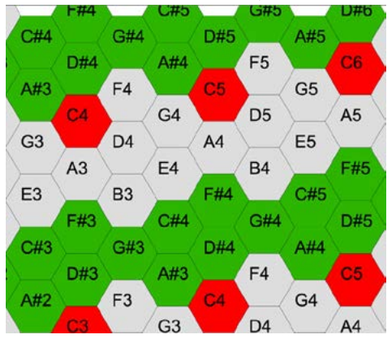
Figure 5: Over-layer design

The basic design of the interface is based on over-layer strategy. There are three layers as shown in Fig. 5: The button layer, which is invisible to users, is used for running a synthesizer on the background. The synthesizer is created by Web Audio API [14], which is able to produce many different musical instrument sounds. The middle layer, which is visible to users, is used for rendering a particular isomorphic layout chosen by users. The top layer, which is invisible to users, is used for bundle listeners to detect users' behaviours (navigation, click and touch). Once listeners detect a click or touch behaviour, animation of the clicked tile will be activated and synthesizer will to called to sound a note.

3.2 Features



(a) Gerhard Layout



(b) Park Layout

Figure 6: Isomorphic layouts rendered in the middle layer of WebHexIso. Root note (C) is red, and notes that would normally be black on a piano are marked in green.

The interface provides options for selecting typical layouts (Harmonic table, Wicki-Hayden, Jankó, etc.), and additionally, users can define their own custom layouts by choosing the musical interval in horizontal and vertical directions.

Users can also switch the direction of the layouts horizontally (the Zigzag direction of hexagonal grid [15] faces north) or vertically (the Armchair direction of hexagonal grid [15] faces north). Users can choose any note for the tonic, and can choose to colour notes based on any scale, key, or mode. The colour of layouts, the size of the keys and the type of synthesizers are manageable.

3.3 Scalable Multi-touch API and Web MIDI API

Beyond the basic design, by using Multi-touch API with touch events functions to control the multi-touch behaviour, the interface provides an opportunity to be used as a mobile application. The WebHexIso and other existing mobile apps such as Musix Pro have similar functionality, when the web-based interface is opened in a modern browser.

Furthermore, the Web MIDI API [16] allows MIDI musical controllers transfer data by USB. Once a musical controller plugged-in while the WebHexIso is open, the WebHexIso will detect and identify the controller and receive MIDI notes data from the controller. Fig. 7 shows when WebHexIso detects the AXiS-49 is plugged-in, the corresponding layout is shown on the screen. Moreover, the Web MIDI API also allows data transfer from WebHexIso to MIDI devices, which allows WebHexIso activates plugged-in slave MIDI device and either play notes on a synthesizer, or send note event and layout patterns to it.

Figure 7: The interface shows harmonic table layout on the screen when AXiS-49 is plugged-in

3.4 Limitation

Depending on the browser and computer being used, WebHexIso may have increased latency. On a modern browser and recent computer, the latency should not be noticeable, but on slower systems, if WebHexIso performs slowly to the point that a noticeable lag is present between the activation of a key and the sounding of a note, it would be possible to disable some features (like multitouch) to increase performance at the cost of functionality.

4. CONCLUSION AND FUTURE WORK

A novel customizable web-based hexagonal isomorphic musical keyboard interface has been introduced. Users can define or select different isomorphic keyboard by themselves. This interface is online and free so that more people could have the chance to access the isomorphic keyboards. By using multi-touch API, the web-based interface can implement behaviours as a mobile application. It also could be used as an assisting screen of isomorphic layouts for performance when a MIDI controller device, such as AXiS-49, is plugged-in.

In future, more MIDI devices will be recognized by WebHexIso when their MIDI names are built into the database. Furthermore, a user-interface study of can be conducted. Based on this system, the composers can find more isomorphic layout patterns, which will benefit performance.

5. REFERENCES

[1] L. Euler, "De Harmoniae Veris Principiis per Speculum Musicum Repraesentatis," in *Novi commentarii academiae scientiarum Petropolitanae*, St. Petersburg, 1774, pp. 330 – 353.

[2] B. Park and D. Gerhard, "Discrete Isomorphic Completeness and a Unified Isomorphic Layout Format," in *Proceedings of the Sound and Music Computing Conference*, Stockholm, Sweden, 2013.

[3] ——, "Rainboard and Musix: Building Dynamic Isomorphic Interfaces," in *Proceedings of 13th International Conference on New Interfaces for Musical Expression*, 05 2013.

[4] A. Milne, A. Xambó, R. Laney, D. Sharp, A. Prechtl, and S. Holland, "Hex Player — a virtual musical controller," in *Proceedings of the 11th International Conference on New Interfaces for Musical Expression(NIME)*, Oslo, Norway, 2011, pp. 244 – 247.

[5] SkyLight and Denryoku, "Hex OSC Full," http://www.sky-light.jp/hex/, Online.

[6] D. Randolph, J. Haigh, and S. Larroque, "Hexiano," https://github.com/lrq3000/hexiano/, Online.

[7] V. Goudard, H. Genevois, and L. Feugere, "On the Playing of Monodic Pitch in Digital Music Instruments," in *Proceedings of the 40th International Computer Music Conference (ICMC) joint with the 11th Sound and Music Computing conference (SMC)*, Athens, Sep 2014, p. 1418.

[8] P. von Jankó, "Neuerung an der unter No 25282 patentirten Kalviatur," German patent 25282-1885.

[9] K. Naragon, "The Jankó Keyboard," *typescript*, pp. 140–142, 1977.

[10] D. Tymoczko, "Geometrical Methods in Recent Music Theory," *MTO: a Journal of the Society for Music Theory*, vol. 1, 2010.

[11] H. Riemann, "Ideen zu einer "Lehre von den Tonvorstellungen"," in *Jahrbuch der Bibliothek Peters*, 1914 – 1915, pp. 21 – 22.

[12] S. Little, "CanvasBoard," https://github.com/uber5001/CanvasKeyboard, online. Accessed on 8-Jan-2016.

[13] K. Wicki, "Tastatur für Musikinstrumente," Swiss patent 13329-1896.

[14] W3C, "Web Audio API," https://www.w3.org/TR/webaudio/, Online. Draft 8-Dec-2015.

[15] H. Hu, B. Park, and D. Gerhard, "On the Musical Oppotunities of Cylindrical Hexagonal Lattices: Mapping Flat Isomorphisms Onto Nanotube Structure," in *Proceedings of the 41th. International Computer Music Conference*, Denton, Texas, 2015, pp. 388–391.

[16] W3C, "Web MIDI API," https://www.w3.org/TR/webmidi/, Online. Draft 17-March-2015.

Roulette: A Customized Circular Sequencer for Generative Music

Daniel McNamara
California Institute of the Arts
danielmcnamara@alum.calarts.edu

Ajay Kapur
California Institute of the Arts
akapur@calarts.edu

ABSTRACT

Roulette is a generative sequencer that uses probability algorithms to create rhythmically varied drum patterns. The intent of Roulette is to evolve the static nature of sequential composition to enable its user to program drum patterns in a dynamic fashion. From the initial design of a rhythm, varied patterns are emitted outlining a principal musical form.

1. INTRODUCTION

There is a long history of electronic musicians using sequencers in compositions and performance. Over the past century, there has been a continued development of sequencers that have been created by industry and individual researchers. Most of the commercially popular sequencers consist of similar design principals and functionalities. With the rising popularity of custom musical software creation there is a plethora of new software sequencers that explore the expanded capabilities of the sequencer as a compositional tool. What is missing from these new soft-sequencers is an interface that properly reflects the interaction possible from the software. Due to the lack of custom hardware most experimental sequencers are ultimately controlled by mapping their parameters to MIDI-controllers; this dilutes the relationship between hardware and software. Cross-referentially designing both the software and the hardware of an electronic instrument is a holistic approach that aims to underline its individual character and compositional capabilities.

In this paper the entire creation process of Roulette is described. Section 2 discusses inspiration and related work, while section 3 discusses design and implementation. Finally, section 4 discusses custom software built for the Roulette.

2. INSPIRATION AND RELATED WORK

This section discusses artists that have inspired this work in three areas: the use of the circle as a compositional tool, customized sequencer design, and generative sequencers.

2.1 The Circle as a Compositional Tool

The circle is used metaphorically to conceptualize abstract concepts such as repetition, perfection and/or imperfection, and the antithesis of rigidity. Musically a circle illustrates the exact idea of a sequence, but in common music notation, software, and hardware a sequence is designed within the rigidly of a square-influenced interface.

The circle is a common point of influence in regards to reconsiderations to the paradigms of electronic music composition. Dan Trueman's *Cyclotron* [1] was designed in the pursuit of composing electronic music in the Norwegian Telemark style, a form of music that varies rhythmically in ways that cannot be achieved through standard tempo subdivision. Adam Places' *AlphaSphere* [2] was made as an investigation into the design and aesthesis of contemporary nimes; the *AlphaSphere* has a very prominent circular design as a way of investigating new and original modes of interaction. Trimpin's *Sheng High*, Daniel Gábana Arellano's Radear [3] and Spencer Kiser's *spinCycle* [4] both utilize the circle's inherent looping mechanic in order to create physical sequencers.

2.2 Customized Sequencer Design

In Rafael Arar's paper outlining the history of sequencers he traces their lineage of development [5]. The paper ends by examining the current design paradigm of sequencers as well as showcasing contemporary experimental configurations. The current paradigm of commercial sequencer design is the grid-based model, originating from the Monome[1] and subsequently implemented in commercial products such as Ableton Push[2] and Novation Launch Pad[3]. The grid-based model is a hyper-rigid design comprising of a square with a nested matrix of squares. Although there is no doubt potential for expression, the grid-based model echoes the compromises that must considered when designing for industrial production methods. In contrast to the current paradigm of commercial sequencers, Arar's paper also examines independently developed sequencers highlighting the potential for alternative experiences and interaction available from sequencers that aren't typically utilized.

[1] http://monome.org/
[2] https://www.ableton.com/en/push/
[3] https://www.ableton.com/en/products/controllers/launchpad/

2.3 Generative Sequencing

The idea of creating a sequence that is controlling thematic content as opposed to note specific content is also explored in Marcelo Mortensen Wanderley and Nicola Orio's paper on musical interaction devices [6]. This paper proposes contexts of musical control. It first defines the concept of *note-level-control*; a one to one interaction with an instrument. It then defines *score-level-control*, the idea of controlling music in a stance similar to a conductors level of interaction.

Luisa Pereira Hors' *Well Tempered Sequencer* [7] explores a series of generative sequencers that "create music in dialog with the user, who is given varying degrees of control over the system."

Combining knowledge learned from all of the research described in this section were key inspirations in designing and creating Roulette.

3. DESIGN AND IMPLEMENTATION

This section discusses the goals and design principals explored in Roulette's creation as well as how these concepts were physically executed.

3.1 Core Design Concept

Roulette's design processes focused on the implementation of a circular aesthetic. This decision was made in order to achieve an alternative user experience; one that strays from the typical experience derived from the square-centric designs that are prominent in commercially available sequencers. The circle aesthetic also reinforces the conceptual value of a generative sequencer. As a circle holds a specific shape but is void of points, Roulette has a musical form but is void of specificity.

Wherever possible, circles were implemented into the design they are a prominent feature in the ring-formed body, the knobs and buttons, and the combined rotary soft-pot and encoder in the centre of the sequencer that creates the concentric-circle master control module.

Roulette comprises of a specific module design that is evenly distributed in 16 steps around the ring of the sequencer.

3.2 The Module and Its Controls

Each module consists of a number of components for sequence development. An orange 3mm LED indicates which step the sequencer is currently on. A blue LED button indicates whether or not a drum hit is going to occur; the button gives the user the ability to cancel or initiate a drum hit event. A thumb-slider joystick that has been modified to have no spring recoil acts as a two-dimensional pot for setting two parameters: velocity range, and event probability odds. Finally the rotary potentiometer at the top of the module controls the timing offset of the drum hit event; execution of the event can occur before, at, or after the proper 16th note division.

All module components that define a range function by recording their position to two separate variables. In order to set each side of the range there is a shift button functionality assigned to the push button on the encoder located in the central module, depending on whether or not the shift button is pressed the change in position of the sensor is recorded as its ranges lowest or highest point.

3.3 Module PCB

Due to the size, scale of production, and the circular form of Roulette special considerations were required in the design of its circuitry. Printing circuit boards was a necessity, as the compact form factor of roulette would not physically allow hand-soldered perforated boards to fit.

With modularity in mind, an independent PCB was printed for each module instead of creating one PCB for the entire instrument. The modular PCB approach benefits the design in two ways. First, modules offer some leeway for a reconfiguration of physical placement during the design process if a change is necessary. The second benefit to modules is that they greatly reduce the price of production; this is because the pricing of PCB production is based on cubic material usage. Printing modules insures that PCBs only occupy the absolute amount space they need to (see Figure 1).

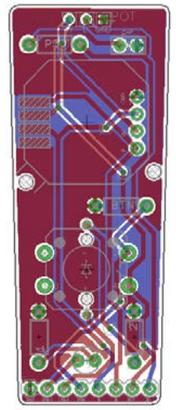


Figure 1. A rendering of an independent module's PCB with optimal form factor considered.

All 16 modules are wired to a central shield consisting of multiplexers that parse the sensor data accordingly. Multiplexers are required due to the amount of sensors implemented surpasses the available inputs on an Arduino Mega, the microcontroller that Roulette uses.

3.4 The Central Module

The central module controls global Roulette's settings. These settings include: Volume, Tempo, Track Selection, and Shift Mode. It resides in the centre of Roulette following a concentric-circle layout; first there is a rotary

soft-pot and then centred within is a push-button-enabled encoder.

The push button of the encoder controls the shift setting of Roulette. This allows each sensor the capability of recording its data to two different variables. In regards to the central module's data, the shift key switches the rotary soft-pot from track selection to setting tempo. Finally the encoder knob sets the master volume level of the instrument.

3.5 Fabrication Techniques

Multiple fabrication techniques were applied in the build process of Roulette in order to achieve the project's aesthetic goals while working within the monetary and time-based restrictions inherent in an independent project. The following subsections list the project's core fabrication techniques.

3.5.1 CAD Modeling

CAD modelling software was a critical tool from initial planning and fully through to the end of development (see Figure 2). The use of CAD software was infinitely helpful as once ideas were fully developed, those same files could be used without alteration in the process of physically creating them.

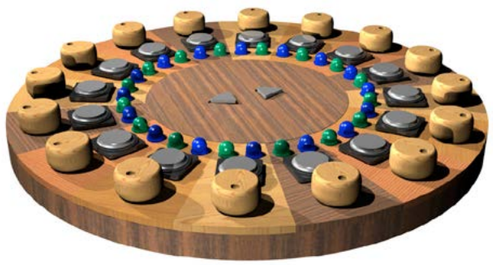


Figure 2. The initial rendering of Roulette illustrating its principle design concepts.

3.5.2 Paper Prototyping

Paper prototyping helped in the task of deciding on the module layout and sequencer size that felt the most comfortable. From surveying multiple sizes printed to scale on paper, a 10-inch diameter for Roulette was decided upon as the most comfortable and practical scale for the build (see Figure 3). Paper prototyping was integral in order to save time and money before pursuing a full build.

Figure 3. A paper prototype printed to test the distribution scale per module on a 10-inch faceplate.

3.5.3 CNC Machining

After successful paper prototyping, CNC machining was used to physically produce the faceplate and centre module of Roulette (see Figure 4). The geometrically complex pattern of Roulette's faceplate would have proven highly difficult to execute with standard shop-tools; the use of rapid prototyping technology enables aesthetic exploration of the tools typically used in commercial product design.

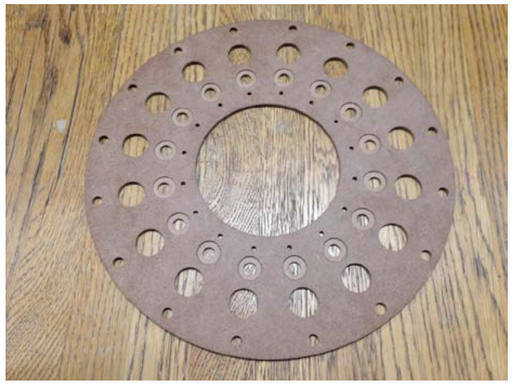


Figure 4. The 10-inch faceplate cut from a CNC machine after passing the paper prototyping approval phase .

3.5.4 Prefabricated Materials

As Roulette required a cylindrical body a consideration of the most appropriate fabrication method to be used had to be made. The initial fabrication solutions to this were either layering rings cut with a CNC machine, or steaming veneers. Both solutions were less than ideal as they would be both expensive and time consuming. Especially in the case of steaming wood, as it would require a highly specific skill. Both initial ideas were dismissed after the realization that a repurposed drum shell would meet the criteria perfectly.

In addition to the structural requirements being met, an aesthetic harmony is achieved by drawing a connection between physical drums and Roulette's role as a drum sequencer.

4. CUSTOM SOFTWARE

This section discusses the software written in order for Roulette to sequence in a probability-based fashion, its graphical user interface, and its Arduino-based communication architecture.

4.1 ChucK

The core functionality of Roulette is written in the ChucK programming language[4]. ChucK was chosen as its time-based functionality made it the most appropriate and effective in programming a sequencer.

Roulette's sequencer program consists of 16 independent instances of a module class. Each module class receives data from the users settings on the Roulette interface. The sensor data determines how the probability functions within each independent object will function. Ultimately the output from these functions is whether or not a drum hit will occur, and if so when in relation to its quantized note division it will occur.

4.2 Processing GUI

A complimentary GUI for Roulette was designed in order to supply ample user feedback (see Figure 5). The GUI was built with the Processing[5] programming language and receives OSC data via ChucK about the state of all sensors and settings on the sequencer. Users have the option to manipulate Roulette from either its physical interface or the GUI.

4.3 Arduino

The Arduino[6] microcontroller is responsible for all sensor data parsing. A collection of multiplexers collect all sensor data, then afterwards the software written on the Arduino uses a delta comparison system to only parse data over serial to ChucK if it detects a change in state. This saves energy greatly and makes it easier on the ChucK side to deal with incoming information.

5. CONCLUSION

Roulette offers expanded capabilities for sequencing drum patterns; specifically it offers a way to create dynamic drum patterns. Although a circular shape is not commonplace in interface design, conceptually it is highly referential to typical conceptions of time. This allows for reconsiderations of drum pattern creation. In future implementations a slip ring will be implemented to allow Roulette to spin, allowing users a greater amount of ease when altering module settings.

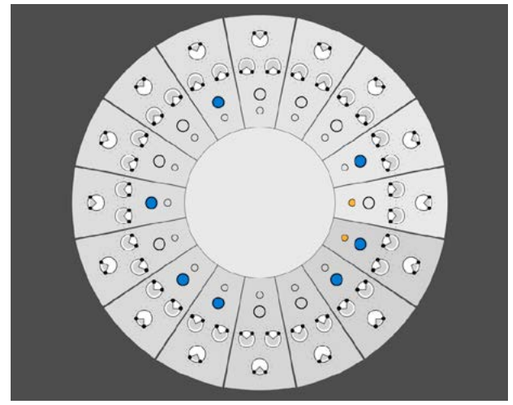


Figure 5. Roulette's GUI built in Processing. All knobs and buttons are fully interactive and can manipulate the ChucK software.

[4] http://chuck.cs.princeton.edu/
[5] https://processing.org/
[6] https://www.arduino.cc/

6. REFERENCES

[1] D. Trueman, "The Cyclotron: a Tool for Playing with Time," in *Proceedings of the International Computer Music Conference*, 2007.

[2] A. Place, L. Lacey, and T. Mitchell, "AlphaSphere from Prototype to Product.," in *NIME*, 2014, pp. 399–402.

[3] D. G. Arellano and A. McPherson, "Radear: A Tangible Spinning Music Sequencer.," in *NIME*, 2014, pp. 84–85.

[4] S. Kiser, "spinCycle: a color-tracking turntable sequencer," in *Proceedings of the 2006 conference on New interfaces for musical expression*, 2006, pp. 75–76.

[5] R. Arar and A. Kapur, "A HISTORY OF SEQUENCERS: INTERFACES FOR ORGANIZING PATTERN-BASED MUSIC," 2013.

[6] M. M. Wanderley and N. Orio, "Evaluation of input devices for musical expression: Borrowing tools from hci," *Computer Music Journal*, vol. 26, no. 3, pp. 62–76, 2002.

[7] L. P. Hors, "The Well–Sequenced Synthesizer: a Series of Generative Sequencers."

MusicBox: creating a musical space with mobile devices

Jan-Torsten Milde
Fulda University of Applied Sciences
milde@hs-fulda.de

ABSTRACT

This paper describes the ongoing development of a system for the creation of a distributed musical space: the MusicBox. The MusicBox has been realized as an open access point for mobile devices. It provides a musical web application enabling the musician to distribute audio events onto the connected mobile devices and control synchronous playback of these events.

In order to locate the mobile devices, a microphone array has been developed, allowing to automatically identify sound direction of the connected mobile devices. This makes it possible to control the position of audio events in the musical space.

The system has been implemented on a Raspberry Pi, making it very cheap and robust. No network access is needed to run the MusicBox, turning it into a versatile tool to setup interactive distributed music installations.

1. INTRODUCTION

Modern web technology can be used to create highly interactive, visually appealing, collaborative creative applications. With the development of the Web Audio API, a promising basis for the creation of distributed audio application has been made available. A number of interesting projects have shown some progress in bringing musical application into the web browser: Flocking ([1]) defines a framework for declarative music making, Gibber allows for live coding in the browser ([2]) and Roberts et.al. ([3]) show how the web browser could be used as basis for developing synthesizers with innovative interfaces.

In our approach, we would like to use the web technology to create musical spaces, thus distributing the sound creation onto a number of small mobile devices placed in a large room. These devices should be synchronized in time and should be controlled by a central musical system: the MusicBox.

2. MUSICBOX: CREATING AN OPEN MUSICAL SPACE

The development and construction of the MusicBox has been driven by the idea to create a music system, which supports the musician to easily define and setup a distributed musical space. The underlying concept is based on a client / server approach, where standard mobile systems (aka smart phones) are used to perform the sound synthesis, or simply the playback of pre recorded sound files. The musician is defining a digital orchestra built from mobile devices. The computing power of standard mobile devices has increased significantly during the last few years, making them feasible for sound reproduction. A limiting factor are the inbuilt speakers, though. In order to perform the synchronized audio playback on the devices, web technologies are used. More specifically a web application using the Web Audio API ([4]) has been implemented.

Figure 1. The MusicBox runs on a Raspberry Pi B. The system is configured to work as an open access point. Mobile devices connect to the MusicBox and start the web application with their browser. Once connected, the timing between the mobile device and the MusicBox is synchronized. The mobile device is then part of the musical space.

3. TECHNICAL SETUP OF THE MUSICBOX

The setup of the MusicBox on the Raspberry PI took 5 steps:

- installation of the standard operating system (Rasperian)
- configuration of access point software
- installation of node.js
- development of the music system as a web application in node.js and express
- integration of the microphone array (via USB connected Arduino).

The first step was to install and setup the standard operating system onto the Raspberry Pi. Rasperian is a Debian based Linux clone compiled for the underlying hardware of the Raspberry. In order to make the system more user friendly, we configured the Raspberry to behave like an open access point. A pre compiled version of the *hostap*-demon was installed on the system. This version matched the used WLAN-dongle. Once the access point demon was running, connecting to the box was very simple. In order to make things even simpler, we decided to install *dnsmasq*. With this service symbolic names could be used for identifying the MusicBox. In our case we chose *http://musicbox.fun* as the address of the system. This even works without any connection to the internet. The MusicBox thus provides an independent network, which can be used as a basis for musical installations even in very unusual places.

The musical web application has been implemented in node.js. Node.js is targeting the development of modern web applications. It is well suited for the implementation of JSON-based REST services, supports data streaming and the development of real-time web applications. As such it is matching the requirements of distributed musical application very well. On the other hand, it is not powerful enough for heavy real time computation tasks. So real time audio processing should rather be implemented in other frameworks.

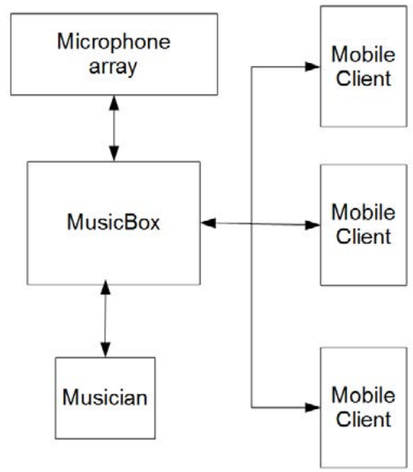


Figure 2. The system architecture of the MusicBox. The web application is synchronizing the mobile devices. It provides a control interface to the musician, that allows to transmit audio data and control data.

The final configuration of the MusicBox prototype has been stored as an iso-image. This makes it very easy to create a running copy of the system, even for beginners with no technical background. Simply create a copy of the iso-image on a standard micro sd card and insert the card into the Raspberry Pi. If the WLAN-hardware matches the standard configuration, the system will be up and running in less then a minute.

4. SYNCHRONIZING DEVICES

An essential pre condition for the implementation of a distributed musical environment is establishing a precice time management for the underlying web application. The basic timing of server (MusicBox) and clients (mobile devices) has to be tightly synchronized. Without this synchronization many musical applications like synthesizers, sequencers, drum machines, loop boxes or audio samplers will not work as expected. Audio events need to be synchronously scheduled in order to create a coordinated playback in an distributed environment.

In order to establish this synchronisation, one could be tempted to use the wireless connection to provide dynamic synchronisation messages. This turns out to be not very reliable. The IP protocol stack is not well suited for real time application communication. It cannot be guaranteed that an IP package is arriving in time.

As a consequence we have chosen to rely on the internal clocks of the mobile devices. This simplifies the synchronisation task and at the same time reduces the networks traffic of the running application.

4.1 Setting the timing offset

Once connected, the mobile devices are synchronizing their system time with the system time of the central MusicBox. The temporal difference between the two timers has to be calculated as precise as possible in order to achieve a high timing correspondence for the nodes of the distributed system.

The synchronization of the system times is a two step process. At first a simple AJAX request is executed upon loading the application (an html page) onto the client. The requests connects to a central web service on the MusicBox and transmits it's system time. The standard resolution is milli seconds.

In a second step a WebSocket connection is established. This connection is used to further refine the adjustment of the time difference between server and client. Our approach follows the solution described in [5]. We adopted the given implementation to the technology used in our setup.

All in all the taken synchronization method results in a quite high precision of the timing adjustments. In our experiments the first phase calculates a temporal difference in the timers of only 30ms. The faster second phase is able to decrease this difference to below 4ms.

As a consequence the deviation of the timing in the complete cluster is approximately 8ms. Starting and stopping audio events on the mobile devices is therefore comparatively synchronous and below the general perceptual threshold.

4.2 Web Audio

The Web Audio API *AudioContext* allows to access the current time (the audio clock) using the *currentTime* property. This property is calculated by accessing the system hardware and returns an increasing double value denoting the number of seconds since the *AudioContext* has been established. The internal resolution of a double value is sufficient to facilitate a very precise timing for the audio events even over a longer period of time. Within the Web Audio API a larger number of functions are controlled by the audio clock. As such it becomes possible to precisely control the timing of the audio events with this property.

For the current version of the MusicBox we rely on this relatively simple form of timing control. It is sufficient to achieve a synchronous start of audio playback across the mobile devices of the musical space.

A clear drawback of this approach is the strict fixation of the timing. Once set, it is not possible to dynamically adjust audio parameters with this simple timing approach. More elaborate timing control for WebAudio applications have been discussed by Wilson ([6]) or Schnell et. al. ([7]).

5. MICROPHONE ARRAY

As soon as the mobile devices have been registered and synchronized, the audio playback can be started. At this point in time no further information about the spacial placement of the mobile devices is available. One could think about using the GPS sensor to request the current position. Unfortunately the precision of this sensor is not very high and quite often the sensor does not work inside buildings.

As a consequence, we developed a simple microphone array to at least achieve a rough estimation of the relative position of the mobile devices. This microphone array is able to detect the relative volume of a mobile device and assigns this to a sector in space. The microphone array consists of 6 identical microphones with a pre-amplifier, that are arranged in a semi circle of 180 degrees with each microphone being responsible for a sector of 30 degrees. It can be mounted on a standard tripod.

The analog to digital conversion is performed by an Arduino, that is connected to MusicBox via USB. The interpretation of the measurements is implemented as part of the web application. Despite of the fact, that the microphone array is definitely a low cost, low tec solution, the resulting sector assignment works surprisingly good.

6. WEB APPLICATION WITH NODE.JS

The musical web application was developed using Node.js. The following services have been implemented in the web application:

- for the initial synchronization a *web service* has been implemented providing the current system time in ms
- a *web socket* based connection to the client is established for sending timing information and control data
- a *visualization* show the attached clients and also displays status information of the clients
- *transmission of audio data* to the clients
- *transmission of control data* as part of the play back control

The web application is split up into two principle application parts: the client part for the mobile devices and the control part for the musician.

The graphical user interface for the client part is kept very simple. It consist of a single colored area displaying the current system (client) status using a traffic light metaphor. The area lights up in green when the client is synchronized and ready to play back audio content. If the area is yellow, audio data is uploaded to the client. The client is in waiting state. And if it lights up in red, then synchronization has failed. The client is not operational.

The control part for the musician is realized by a separate user interface. Once the clients are synchronized, the musician is able to send commands and audio data.

The play back control is based on the visualization. It is comparable to a traditional score notation or the instrumental track display that can be found in most of the current digital audio workstations. The audio elements are assigned to the clients via drag 'n drop (alternatively by a double click). Each audio element has flexible but fixed duration, that is determined by the underlying temporal grid. The musician places the audio element at it's intended temporal position, while the actual play back of an audio event is initiated by sending the starting time to the client.

6.1 Recording sound on the MusicBox

In order to make the MusicBox more flexible and better suited for live performances, we added a USB audio interface to the system. This allows to record audio synchronously with the ongoing performance. The MusicBox thus works like simple loop station. In addition, a set of effects has been implemented (*delay, rever, filter etc.*).

The recorded audio data can then be streamed to mobile devices and then be integrated into the ongoing playback. The musician is therefore able to create audio events on the fly, distribute them to the mobile devices and integrate these audio events into the musical space.

7. CONCLUSIONS

The MusicBox provides a simple, yet robust and flexible environment to easily create a distributed musical space using standard mobile devices. It synchronizes audio playback across the cluster and simplifies the spacial positioning of the connected audio clients.

The current setup is a good basis for further investigations into the creation of musical spaces.

Until now, the clients are passive. While it is possible for multiple musicians to interact with MusicBox simultaneously, no human interaction can be executed by the clients. It would be interesting to extend the functionality of the mobile device into this direction.

While we are currently using web technology, the general approach is not limited to this approach. The MusicBox could quite as well serve native applications on the various devices. These could be more powerful audio applications like software synthesizers that could considerably expand the expressiveness of the musical space.

Acknowledgments

8. REFERENCES

[1] C. Clark and A. Tindale, "Flocking: a framework for declarative music-making on the Web," in *Proceedings of the 2014 International Computer Music Conference*, Athen, 2014.

[2] J. K.-M. C. Roberts, "Gibber: Live Coding Audio in the Browser," in *Proceedings of the 2012 International Computer Music Conference*, 2012.

[3] C. Roberts, G. Wakefield, and M. Wright, "The Web Browser as Synthesizer and Interface," in *New Interfaces for Musical Expression conference*, 2013.

[4] C. W. P. Adenot and C. Rogers, "Web Audio API," in *http://webaudio.github.io/web-audio-api/*, 2016.

[5] N. Sardo, "GoTime," in *https://github.com/nicksardo/GoTime*, 2014.

[6] C. Wilson, "A Tale of Two Clocks - Scheduling Web Audio with Precision," in *http://www.html5rocks.com/en/tutorials/audio/scheduling/*, 2013.

[7] N. Schnell, V. Saiz, K. Barkati, and S. Goldszmidt, "Of Time Engines and Masters: An API for Scheduling and Synchronizing the Generation and Playback of Event Sequences and Media Streams for the Web Audio API," in *Proceedings of the 1st annual Web Audio Conference*, 2015.

Frequency Domain Spatial Mapping with the LEAP Motion Controller

David Kim-Boyle
University of Sydney
david.kim-boyle@sydney.edu.au

ABSTRACT

This paper explores the use of a LEAP Motion Controller for real-time control of frequency domain sound spatialization. The principles of spectral spatialization are outlined and various issues and solutions related to the efficient and usable control of large data sets within the MaxMSP/Jitter environment are presented. The LEAP Motion Controller is shown to offer a particularly elegant means through which useful control of various mapping and distribution techniques of spatial data may be driven during live performance.

1. INTRODUCTION

The LEAP Motion Controller is a relatively newly developed infra-red sensor that opens up exciting new possibilities for interacting with sonic data in real-time. The author has found it to be a particularly useful device for the real-time manipulation of data associated with frequency domain spatialization. The real-time control of various mapping, distribution, and transformation strategies of such large data sets presents distinct challenges in live performance environments for which the LEAP Motion Controller affords an elegant solution.

While various real-time methods of sound spatialization such as IRCAM's SPAT [1] or ZKM's *Zirkonium* [2] and associated techniques of gestural control [3, 4] have evolved over the past twenty years, the majority of these techniques have been developed to transform the spatial mapping of complete sonic objects, gestalts or collectives [4, 5, 6]. With frequency domain spatialization, however, it is desirable to have independent control over discrete bands of spectral information. This presents immediate challenges of both a computational and psychoacoustic nature. On the computational and control level, discrete spatial mapping of spectral data requires control of large data sets which cannot be effectively realized at a high-level resolution with most gestural interfaces. On the psychoacoustic level, spatial segregation of spectral bands is highly dependent on the timbral quality of the source sound. So a mapping technique that is effective for one particular type of sound may deliver quite different results when applied to another [7, 8].

Mindful of these considerations the author has explored various methods of spatially mapping spectral data including the use of particle systems and boids [9]. While these techniques have provided results rich in musical possibility, the ability to drive mapping and distribution techniques through physical gestures captured through devices such as the LEAP Motion Controller, considerably extends the live performance capacity of the technique.

2. FREQUENCY DOMAIN SPATIALIZATION

Unlike traditional techniques of sound spatialization in which the full spectral content of a sonic object is mapped to one or many point sources [4], in frequency-domain spatialization, the spectral data contained in the individual bins of an FFT analysis are mapped to discrete spatial locations upon resynthesis, see Table 1 [10]. Such a technique can allow complex relationships between a sound's timbral quality and its spatial dispersion to be explored. For example, through algorithms designed for the control of complex systems [9], the timbral content of a sound can be spatially smeared, focused, or made to disperse around the listener in cloud-like formations but this is, as previously noted, somewhat contingent on the harmonic content of the sound source.

	Left	Center	Right	LS	RS
Sound A	√	√	√	-	-
Sound B	-	-	-	√	√
Sound C	-	√	-	-	-

	Left	Center	Right	LS	RS
FFT Bin 0	√	-	-	-	-
FFT Bin 1	-	√	-	-	-
FFT Bin 2	-	-	√	-	-

Table 1. Overview of traditional spatialization with three sound sources (A, B, C) panned to discrete loudspeakers (upper), and spectral spatialization with spectral data contained in the first three FFT bins mapped to discrete loudspeakers (lower) in a 5.1 system.

2.1 MaxMSP Implementation

The frequency domain spatialization is implemented in MaxMSP. Employing a process similar to that of Lippe and Torchia [11], an FFT analysis is performed on a signal and the spectral data stored in the FFT bins is distributed across eight output channels upon resynthesis of the frame. The routing matrix of spectral data is read from a two channel signal buffer indexed within the FFT. This buffer is itself determined by a two-plane x/y jitter matrix controlled from outside the FFT through various preset distributions and mappings, and by the LEAP Motion Controller. While it is a relatively straightforward process to extend the spatialization to three dimensions, through the use of three rather than two-plane matrices, the author has chosen not to do so given the relative sparsity of diffusion systems which include loudspeakers mounted along three axes. The audio outputs from the resynthesized FFT frame are routed through Jan Schacher and Philippe Kocher's MaxMSP ambisonic externals [12], which provide an additional layer of output control. A schematic of the process is shown in Figure 1.

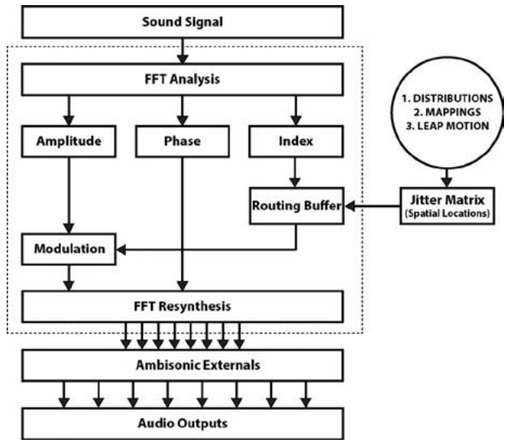


Figure 1. Frequency domain spatialization. The FFT analysis/resynthesis is contained within a dotted border.

While spectral spatialization offers a rich palette of sonic possibilities and opportunities for the development of new spectral transformations, one of the challenges presented by the technique is how large amounts of data can be meaningfully controlled, especially in live-performance contexts. To this end, the author has explored the use of complex multi-agent systems such as the *boids* algorithm [9], which allows a small number of control parameters such as *separation* (which places constraints on the distance between individual boids), *maximum velocity*, *gravity points* (which establishes points of attraction to which the flock moves), and *inertia* (which affects the boids initial resistance to directional change) to drive the spatial mapping of spectral data contained within FFT bins. The degree of precision over spatial mapping and distributions relinquished by more global control parameters such as these is more than compensated for by the efficiency of control.

From these various experiments, a small subset of musically useful spatial mappings and distributions have emerged. In the author's MaxMSP implementation, each of these mappings and distributions are generated through the use of jitter matrices and various matrix transformations. The use of matrix objects for the storage and transformation of FFT spatial data is an intuitive and efficient method for processing large collections of data and the ease with which such matrices can be visualized in the OpenGL environment, is particularly helpful for visual correlation. Six distributions (which determine the spatial location of an FFT bins) and four mappings (which determine which bins are mapped to those locations) have proven especially useful.

2.2 FFT Spatial Distributions

In the author's MaxMSP implementation, spatial distributions of spectral data include – a) Line – a linear distribution of FFT bins along the x-axis; b) Grid – a rectilinear grid with the number of columns able to be independently assigned; c) Circle – a circular distribution of FFT bins around the x/y origin; d) Drift – a dynamic distribution of FFT bins whereby points randomly drift around the x/y plane; e) Boids – a dynamic distribution of FFT bins where the movement of points simulates flock-like behavior; f) Physical – a dynamic distribution of FFT bins modeled on the jit.phys.multiple object which establishes points of attraction within a rectilinear space. Screen snapshots from a rectilinear and circular distribution are shown in Figure 2 with the spatial location of spectral data contained in each FFT bin visually represented by a small colored node.

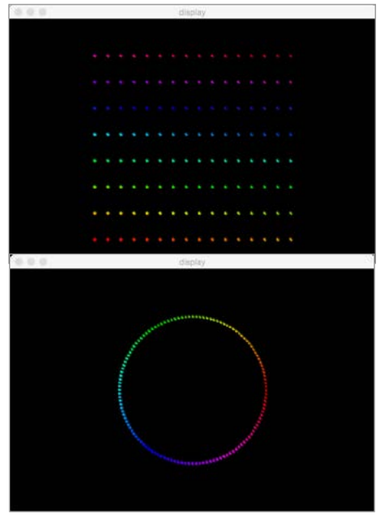


Figure 2. Grid (top) and Circular (bottom) spatial distributions.

2.3 FFT Mappings

Complementing these six spatial distributions are a set of four mappings – a) Linear+ – a linear mapping of FFT bins from bin 0 to FFT window size. If the bins are linearly distributed, this will result in lower to higher bins being spatially mapped from left to right respectively; b) Linear- - similar to (a) but with bins mapped in reverse order from FFT window size to 0; c) Random – a random distribution of bins across points determined by the current distribution; d) Multiplexed – an ordered distribution of bins where even-numbered bins are progressively mapped from left to right and odd-numbered bins are progressively mapped from right to left. The author has found this mapping especially useful as it tends to more evenly distribute spectral energy across a spatial plane for most acoustic sound sources. To facilitate visual recognition of mappings, a simple RED-ORANGE-YELLOW-GREEN-BLUE-INDIGO-VIOLET color scheme has been used, see Figure 2, where lower bin numbers are mapped to red-colored nodes and the highest bins to indigo-colored nodes, irrespective of the magnitude of an FFT bin's amplitude. Linear+ and Multiplexed mappings are illustrated in Figure 3, although for the purpose of clarity, the associated color mappings are not reproduced.

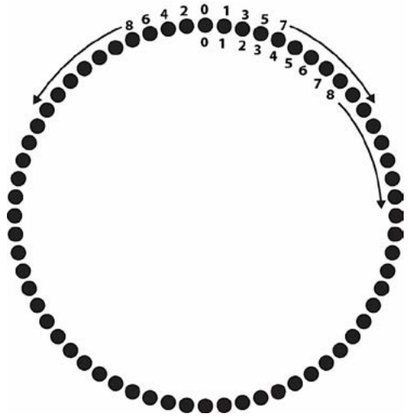


Figure 3. FFT bin mappings – Linear + (inner) and Multiplexed (outer) across a circular distribution.

3. LEAP MOTION CONTROLLER MAPPING

The LEAP Motion Controller uses a dual infra-red sensor to track the movements of the fingers and palms in 3D space. While it has largely been developed and marketed as a controller for virtual reality environments, its application as a creative instrument and musical controller is receiving growing investigation [13, 14, 15, 16]. Diatkine, Bertet, and Ortiz have also explored its use for sound spatialization in 3D space [17]. With its low latency, exceptional responsiveness and accuracy, the LEAP Motion Controller offers a particularly attractive solution to the challenges raised by the real-time control of frequency domain spatialization. Through the monitoring of palm and finger position, obtained through IRCAM's leapmotion external [18] the author has developed a number of intuitive and highly practical methods of frequency domain spatial control.

IRCAM's leapmotion external is able to report on the movement of palms and fingers of both hands simultaneously. Taking advantage of this ability, the author has assigned the right hand as a controller of various spatial properties of FFT bins while data obtained from the left hand is used to control various dynamic properties of the FFT bins as well as to apply certain virtual forces to the spatial trajectories to which the bins are mapped. The basic taxonomy of gestural control is similar to that outlined by Schacher in his approach to gestural control of sound in periphonic space [19], see Figure 4.

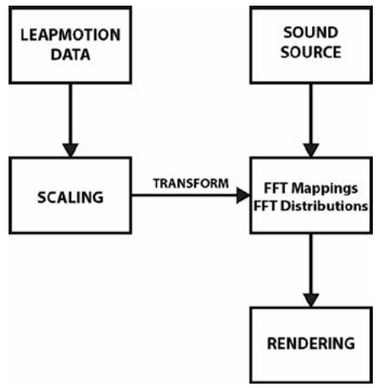


Figure 4. Control schematic of gestural mapping.

The span of the fingers of the right hand provide an effective way of constraining the spatial dispersion of spectral data contained in FFT bins. The fingers become conceptually akin to boundaries of a container within which the harmonic components of a sound source move. When all fingers are brought together, the spatial distribution of FFT bins becomes more constrained as the bins are spatially distributed to one point source. As the fingers are spread apart, the spatial dispersion of spectral data is distributed over a greater area and the timbre of the sound is smeared, according to the chosen pattern of distribution, across a larger spatial field, see Figure 5.

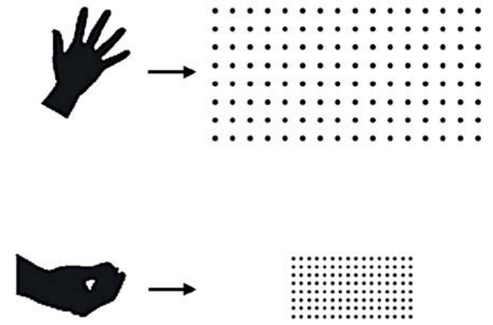


Figure 5. Control of spatial dispersion through varying the separation between the fingers of the right-hand.

The fingertips of the right-hand control the spatial dispersion of spectral data for all distribution patterns other than *Physical*. With the *Physical* distribution selected, the fingertips manipulate the x/y positions of five virtual squares which correspond to points-of-force away from which the spatial locations of spectral data will move. The force enacted at each of these points is determined by the relative fingertip positions of the left hand.

Independent motion tracking of the right-hand palm provides a useful way of driving global changes across the spatial location of all FFT bins. This is very simply achieved in the patch implementation by adding variable offsets, through the *jit.op* object, to the x/y positions stored within the jitter position matrix, see Figure 6.

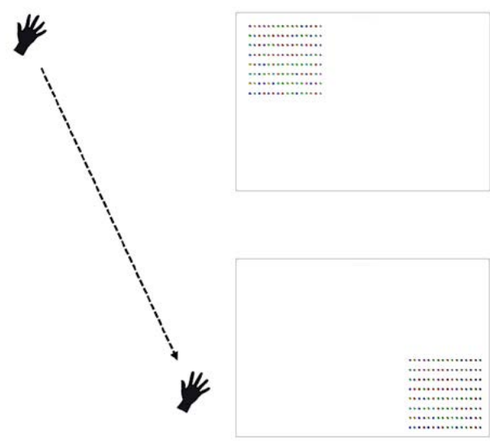


Figure 6. Global changes to a grid distribution of FFT bins through movement of the right-hand palm from top-left to bottom-right.

If the *Boids* distribution is selected, the right-hand palm becomes a point of attraction for all boids within the flock rather than a general panning control for all FFT bins.

Other right-hand data tracked from the LEAP Motion Controller includes both the speed of fingertip changes and rotation of the hand. The former set of data is used to apply a low-level of noise to the spatial location of FFT bins. With fast fingertip twitching, the spatial location of spectral data is rapidly vibrated, creating a shimmering within the spatial field. The rotation of the hand is used to progressively remap FFT bins within the three static distributions (line, grid, circle), giving their resultant timbres a dynamic quality, see Figure 7.

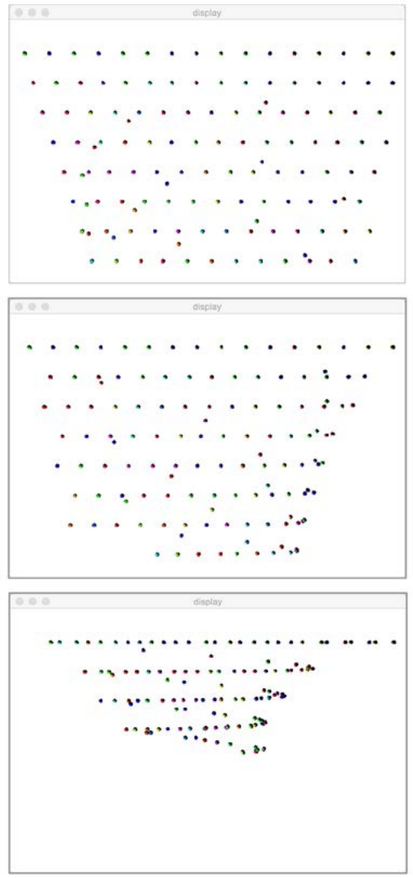


Figure 7. Progressive remapping of a grid distribution through rotation of the right hand.

Given the volume of spatial information able to be controlled by the right hand, the mapping of LEAP Motion data from the left hand has deliberately been kept simple. This has the additional advantage of freeing the hand for controlling other basic operations either within a Max patch or at a mixing console. Other than the assignment of relative force for the *Physical* distributions, the only

parameter tracked is the position of the palm which is used to attenuate the global amplitude of all FFT bins.

The mapping of all data obtained by the leapmotion external is summarized in Table 2.

Distribution	RH Fingers	RH Finger Twitching	RH Rotation	RH Palm
Line	Spatial dispersion	Rapid oscillation of bin positions	Bin re-mapping	Global panning
Grid	Spatial dispersion	Rapid oscillation of bin positions	Bin re-mapping	Global panning
Circle	Spatial dispersion	Rapid oscillation of bin positions	Bin re-mapping	Global panning
Drift	Spatial dispersion	Rapid oscillation of bin positions	NA	Global panning
Boids	Spatial dispersion	Rapid oscillation of bin positions	NA	Flock point-of-attraction
Physical	Points-of-force	Rapid oscillation of bin positions	NA	Global panning

Distribution	LH Fingers	LH Palm
Line	NA	Global amplitude attenuation
Grid	NA	Global amplitude attenuation
Circle	NA	Global amplitude attenuation
Drift	NA	Global amplitude attenuation
Boids	NA	Global amplitude attenuation
Physical	Points-of-force strength	Global amplitude attenuation

Table 2. Mapping of LEAP Motion Controller data to FFT parameters.

4. CONCLUSION

Musically useful control over frequency domain spatialization in live performance environments has always presented a challenge given the large amount of data presented and the equally large range of possibilities for transforming that data. The LEAP Motion Controller presents a viable and low-cost solution although the author acknowledges that a fuller evaluation needs to be conducted. While the use of spectral spatialization has received limited application in the author's creative work, it is a technique that is receiving growing investigation in the work of other composers. The ability to be able to explore these techniques in real-time performance through off-the-shelf devices such as the LEAP Motion Controller will likely add to further explorations of its creative potential.

5. REFERENCES

[1] T. Carpentier, M. Noisternig, and O. Warusfel, "Twenty Years of Ircam Spat: Looking Back, Looking Forward," in *Proceedings of the ICMC 2015 Conference*, Denton, TX, 2015, pp. 270-277.

[2] C. Ramakrishnan, J. Goßmann, and L. Brümmer, "The ZKM Klangdom," in *Proceedings of the 2006 Conference on New Interfaces for Musical Expression (NIME 2006)*, Paris, 2006, pp. 140-143.

[3] M. T. Marshall, J. Malloch, and M. M. Wanderley, "Gesture Control of Sound Spatialization for Live Musical Performance," in *Gesture-Based Human-Computer Interaction and Simulation – 7th International Gesture Workshop*, Berlin: Springer-Verlag, 2007, pp. 227-238.

[4] A. Perez-Lopez, "3DJ: A Supercollider Framework for Real-Time Sound Spatialization," in *Proceedings of the 21st International Conference on Auditory Display (ICAD 2015)*, Graz, 2015, pp. 165-172.

[5] T. Wishart, *On Sonic Art*. Simon Emmerson, Ed. Amsterdam: Harwood Academic Publishers, 1996.

[6] D. Smalley, "Spectromorphology: Explaining Sound-Shapes," in *Organised Sound*, Vol. 2, No. 2, 1997, pp. 107-126.

[7] A. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: MIT Press, 1994.

[8] C. Camier, F.-X. Féron, J. Boissinot, and C. Guastavino, "Tracking Moving Sounds: Perception of Spatial Figures," in *Proceedings of the 21st International Conference on Auditory Display (ICAD 2015)*, Graz, 2015, pp. 308-310.

[9] D. Kim-Boyle, "Spectral and Granular Spatialization with Boids," in *Proceedings of the 2006 International Computer Music Conference*, New Orleans, 2006, pp. 139-142.

[10] D. Kim-Boyle, "Spectral Spatialization – An Overview," in *Proceedings of the 2008 International Computer Music Conference*, Belfast, 2008.

[11] C. Lippe and R. Torchia, "Techniques for Multi-Channel Real-Time Spatial Distribution Using Frequency-Domain Processing," in *Proceedings of the 2003 International Computer Music Conference*, Singapore, 2003. pp. 41-44.

[12] J. Schacher and P. Kocher, https://www.zhdk.ch/index.php?id=icst_ambisonicsexternals. Accessed November, 2015.

[13] D. Tormoen, F. Thalmann, and G. Mazzola, "The Composing Hand: Musical Creation with Leap Motion and the BigBang Rubette," in *Proceedings of the 2014 Conference on New Interfaces for Musical Expression (NIME 2014)*, London, 2014, pp. 207-212.

[14] M. Ritter and A. Aska, "Leap Motion As Expressive Gestural Interface," in *Proceedings of the ICMC/SMC 2014 Conference*, Athens, 2014, pp. 659-662.

[15] J. Ratcliffe, "Hand Motion-Controlled Audio Mixing Interface," in *Proceedings of the 2014 Conference on New Interfaces for Musical Expression (NIME 2014)*, London, 2014, pp. 136-139.

[16] L. Hantrakul and K. Kaczmarek, "Implementations of the Leap Motion in sound synthesis, effects modulation and assistive performance tools," in *Proceedings of the ICMC/SMC 2014 Conference*, Athens, 2014, pp. 648-653.

[17] C. Diaktine, S. Bertet, and M. Ortiz, "Towards the Holistic Spatialization of Multiple Sound Sources in 3D, Implementation using Ambisonics to Binaural Technique," in *Proceedings of the 21st International Conference on Auditory Display (ICAD 2015)*, Graz, 2015, pp. 311-312.

[18] IRCAM Leap Motion Skeletal Tracking in Max. <ismm.ircam.fr/leapmotion/>. Accessed June, 2015. 7 November, 2014.

[19] J. C. Schacher, "Gesture Control of Sounds in 3D Space," in *Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME 2007)*, New York, 2007, pp. 358-362.

Zirkonium 3.1 - a toolkit for spatial composition and performance

Chikashi Miyama miyama@zkm.de

Götz Dipper dipper@zkm.de

ZKM | Institute for Music and Acoustics

Ludger Brümmer lb@zkm.de

ABSTRACT

Zirkonium is a set of Mac OSX software tools to aid the composition and live performance of spatial music; the software allows composers to design multiple spatial trajectories with an intuitive GUI and facilitates arranging them in time. According to the provided trajectory information, the actual audio signals can then be rendered in realtime for 2D or 3D loudspeaker systems. For developing the latest version of Zirkonium, we focused on improving the aspects of usability, visualization and live-control capability. Consequently, a number of functionalities, such as parametric trajectory creation, additional grouping modes, and auto-event creation are implemented. Furthermore, ZirkPad, a newly developed iOS application, enables multiple performers to control Zirkonium remotely with a multi-touch interface and spatialize sounds in live performances.

1. INTRODUCTION

The Institute for Music and Acoustics (IMA) at ZKM Karlsruhe, Germany, is dedicated to electroacoustic music, its production, and performances. The heart of the institute is the *ZKM_Klangdom*. It is a 3D surround loudspeaker system, comprising 43 loudspeakers arranged in the form of a hemisphere. A crucial part of the Klangdom project is the 3D spatialization software *Zirkonium*.

Other major spatialization software, notably IRCAM Spat[1] or ambiX[2], places the primary emphasis upon spatial rendering, and most of them are provided as plug-ins or external objects for Max or Pd. By contrast, the main focus of Zirkonium is spatial notation and composition; Zirkonium provides electroacoustic composers with dedicated editors and tools that enables them to write scores of spatial movements precisely, intuitively, and flexibly. In order to pursue this end, Zirkonium is implemented not as a plug-in but as a standalone Mac OSX application.

Although the primal focus of Zirkonium is fixed-media composition, realtime capabilities of the software have been continuously extended from the begining of the development in 2004. Zirkonium can be controlled by other software remotely via OSC. In this way it is possible to employ Zirkonium also for realtime applications. Moreover, the software is able to send "composed" spatial trajectories as OSC messages in realtime. It enables us to employ external software or hardware for the spatial rendering in addition to its internal rendering algorithms.

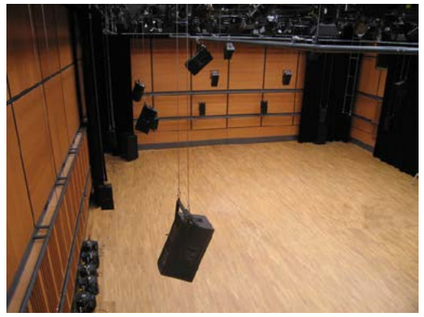


Figure 1. ZKM_Klangdom

For the release of ver. 3.0 in November 2015[3], we improved the aspects of usability, visualization and live-control capability. After the release, multiple additional functionalities were implemented in the software. This paper briefly introduces the overview of Zirkonium ver. 3.0 and new features implemented for ver. 3.1.

2. ZIRKONIUM 3.1

In order to enhance the usability and the visualization, the software structure of the previous version was reassessed and redesigned. The latest version of Zirkonium, version 3.1 consists of three independent applications: *Trajectory editor*, *Speaker setup*, and *ZirkPad*. The *Trajectory editor* is the main application of the software package. It allows to draw and compose spatial trajectories and deliver actual audio signals to the hardware output. The *Speaker Setup* is a utility software that enables the user to define custom speaker arrangements and export them to XML files [1] . The Trajectory editor then imports the XML file and adjusts its spatial rendering algorithms to the speaker arrangement defined in the file. *ZirkPad* is a newly developed iOS App for the release of ver. 3.1. It allows the user to control the Trajectory editor remotely with the multi-touch interface.

2.1 Trajectory editor

The new Trajectory editor provides a superior GUI for designing audio trajectories and arranging spatial events in time (Fig. 2). Unlike the previous version, the Trajectory editor in ver. 3.0 is also responsible for processing actual audio signals from sound files and physical inputs in order to execute spatial rendering for a maximum of 64 loudspeakers.

[1] Refer [3] for the detailed description of the Speaker Setup Application in ver 3.0.

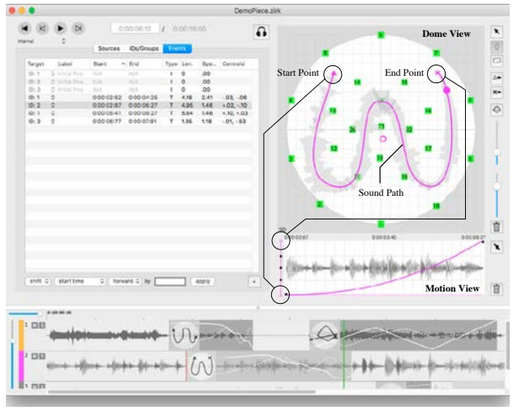


Figure 2. Zirkonium ver. 3.1 Trajectory editor

Figure 3 depicts the software architecture of the Trajectory editor. The Trajectory editor consists of three components: GUI, data management, and spatial rendering engine. For the new version, most of the GUI components are reimplemented with OpenGL and GLSL in order to conserve the CPU resources for the execution of the spatial rendering algorithms. The data management component processes all the data regarding spatial compositions. It has functionalities of importing Speaker-Setup XML files, and exporting spatial events to SpatDIF-XML files [4]. In the spatial rendering engine, the Spatialization server executes the spatial rendering algorithms and distributes the actual audio signals to each output channel. The spatialization server is entirely programmed with Pd (Pure Data)[5] and integrated into the Trajectory editor with the aid of *libPd*, a C Library that turns Pd into an embeddable library. This integration of the Trajectory editor and the Spatialization server does not prevent users from accessing the Spatialization server, running internally in the Trajectory editor. Advanced users with experiences in Pd programming are able to access the patch, modify the core spatial rendering algorithms, and apply arbitrary custom effects (e.g. reverb or doppler) to sound sources. Moreover, thanks to this integration the Trajectory editor is capable of accessing audio content more efficiently than the previous version. The new Trajectory editor offers various new features that take full advantage of this improved efficiency.

The following subsections introduce the most important additional functionalities, implemented in the Trajectory editor.

2.1.1 Trajectory Creation in ver. 3.0

In ver. 3.0, the graphical approach of trajectory creation, inherited from the previous version, is further enhanced.

In the Trajectory editor, a single trajectory (i.e. a movement of a virtual sound source in a specific time frame) is determined by a pair of paths; a *Sound path* and a *Motion path*. These two paths are drawn in two different views: the *Dome view* and the *Motion view*. The Dome view displays a space for spatialization, observed orthographically from

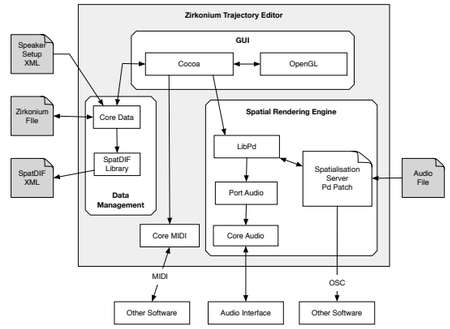


Figure 3. Software architecture of the Trajectory editor

the zenith. In this view, a Sound path, a geometrical route, that a virtual sound source moves along, can be drawn with Bèzier curves. The Motion view and the Motion paths, on the other hand, visualize how a sound source moves along a Sound path in a specific period of time. In ver. 3.0, the Motion path can be drawn with a multi-segment curve. The steepness of each segment is independently configurable by simple mouse operations and it controls the acceleration and deceleration of spatial movements.

Figure 2 shows a possible combination of a Sound path and a Motion path in the Dome view and the Motion view. In the Dome view, a meandering Sound path is defined. A virtual sound source moves along this Sound path from the start point to the end point, marked by a triangle and a cross symbol respectively. The X-axis of the Motion view indicates the time line of an event, and the Y-axis represents the relative position between the start and the end point of the Sound path. The start point (triangle symbol) coincides with the bottom and the end point (cross symbol) coincides with the top of the Motion view. In this way, the Motion view displays the relationship between time and relative position of a sound source, moving along the Sound path. In figure 2, an exponential curve is used as a Motion path. Thus, the sound source accelerates its speed towards the end point (cross). Moreover the waveform of the respective audio file is rendered along the Sound path and behind the Motion path in ver. 3.0. This feature enables users to grasp the relationship between the audio content and its position in space, and to adjust a certain audio content to a specific position in the Dome view.

2.1.2 Parameter-based Trajectory Creation

In addition to the manual drawing method with Bèzier curves, the software provides an algorithmic approach for Sound path creation. By entering a few parameters to the "add circle/spiral" pop-over panel, the software automatically draws a circular or spiral Sound path in the Dome view, employing the minimum amount of Bèzier curves required. These algorithmically drawn Sound paths can be further modified by mouse operations (Fig. 4).

2.1.3 Variety of Grouping modes

We speak of groups in Zirkonium, when several sound objects are moving together. This is a quite efficient way of working, since the movement has to be defined just once

Figure 4. "add circle/spiral" pop-over panel and the result of parameter-based trajectory creation in the Dome view

for the group, and not for each group member separately. The concept of groups has been proven to be very useful especially if some spatial information is already included in the sound material, as it would for example be the case with stereo, quad or 5.0 material. For instance, we might position four sound objects in the form of a virtual quad in Zirkonium, define the quad as a group and move the group in Zirkonium as a whole, elevating or rotating it, while keeping the spatial relationship between the individual group members fixed. Now, if we have movement already included in the original quad source, we will get a superposition of the original movement and the movement created in Zirkonium [6].

It is easy for composers to generate quad or 5.0 source material with other software, since these formats are of course omnipresent and most traditional audio software delivers audio in these or similar formats. Thus, the described approach is a pragmatic way to create spatial complexity in the Klangdom and comparable systems. Another reason for choosing this approach could be the fact that it is not recommended to use more than about 30 sound objects in Zirkonium, partly because of performance issues, partly because of the risk to cause an unnecessary confusion for the user. There are, however, moments, where more than 30 different movements or positions of sounds might be desirable, for example when clouds of sound grains are to be distributed in the space. In these cases, the described approach, possibly using more than one group, yields typically very good results.

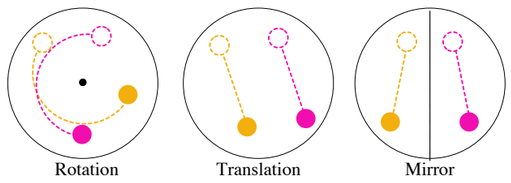


Figure 5. Three Group Modes

The notion of groups has been present in Zirkonium since the very first version. However, the treatment of groups has been considerably refined in ver. 3.1. There are three main group modes, the rotation mode, translation mode and mirror mode (Fig.5). The rotation mode has been available since the early days of Zirkonium. In the rotation mode all sound objects keep the same azimuth and elevation offset among each other. It results in rotational movements around the Z-axis[2], the line between the zenith and the nadir. This is a very effective way of moving a group within the Klangdom – it suites the Klangdom quite well since it has a spherical shape. It is especially suited for surround groups like a virtual quad, because the group stays centered automatically. A special characteristic of the rotation mode is the fact that the spherical distance between the sound objects decreases by increasing their elevation. If, for instance, a virtual quad group is elevated up to the top of the Klangdom, it will eventually end up as a mono source in the zenith. If this is not the desired behavior of a group, the composer can use the translation mode instead. Here the group is moved in parallel translation, which means, that the spherical distance between all sound objects belonging to the group stays always fixed. The third mode, the mirror mode, is especially useful for stereo sources, if their left/right orientation should be kept stable. In the mirror mode, all movements are mirrored against the Klangdom's Y-axis, which is running from the front center to the rear center point.

For the rotation and translation modes there are two minor modes, so we get five different group modes in total. We call the minor modes "fixed" and "free" mode. They define how group movement and individual movement alternate. Generally a group can be deactivated and re-activated again at any moment in a piece. As long as a group is deactivated, its group members can be moved individually as separate sound objects. When the group is re-activated, the members of the group react differently depending on the minor mode. If it is defined as fixed, it regains its initial formation. If it is defined as free, it updates its formation to the current one.

2.1.4 Improved Event handling functionalities

The more a spatial composition project evolves, the more spatial events are used. A powerful event handling tool would be indispensable for increasing productivity. In ver. 3.0, multiple new functionalities are implemented to enhance the efficiency of event handling.

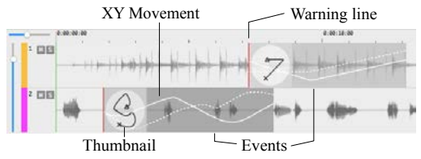


Figure 6. Event view

The Event view is a newly introduced GUI component in ver. 3.0. As shown in figure 6, this view visualizes the waveform of imported sound files and the spatial events assigned to each virtual sound source in the manner of a typical DAW software. In the figure, the waveforms of two sound files are displayed and a few spatial events are represented as transparent dark rectangles, superimposed on the waveforms. On top of these rectangles, snapshots of Sound paths, edited in the Dome view, are shown as thumbnails, and the chronological position of the sound objects is rendered as a solid and a dotted line, either in

[2] Zirkonium adopts the Spherical Navigational System defined in SpatDIF v0.3 specification [7]

cartesian or in polar coordinates. This graphical representation enables the user to quickly grasp the distribution of trajectories in time. Besides, the Trajectory editor also offers the Event filtering panel (Fig. 7). This panel provides a way to automatically select specific events that match particular conditions. All selected events can then be shifted, scaled, copied or erased at once with mouse or keyboard operations.

Figure 7. Event filtering panel

2.1.5 Auto-Event Creation

As the name suggests, this functionality allows the user to create spatial events automatically based on the audio content of imported sound files (Fig. 8). Once this function is executed, the Trajectory editor analyzes the amplitude envelope of the selected sound file and estimates the start and end time of sounds in the file. Based on this estimation, the Trajectory editor creates spatial events for a specific ID or group. This functionality may significantly reduce the amount of time necessary for the event creation by mouse.

Figure 8. Automatically created events based on the amplitude envelope

2.1.6 Other features

For the release of ver 3.0, futher functionalities, such as automatic event interpolation, event filter, reprogramable spatialization server, HOA rendering algorithms, and SpatDIF export functions are implemented. Refer [3] for details.

2.2 ZirkPad

ZirkPad is an iOS application for iPad, that remotely controls the Trajectory editor in realtime and allows users to control the movement of sound objects with its intuitive multi-touch-based user interface.

2.2.1 Motivation

There is a long tradition of sophisticated live-diffusion methods or systems, notably Acousmonium[8] or BEAST[9]. Most of these approaches are primarily specialized in the diffusion of a stereo signal to a large number of loudspeakers. However, they are not perfectly suited to the diffusion of a larger number of input channels. ZirkPad attempts to overcome this limitation by utilizing Zirkonium's traditional "object-based" approach, which gives the performer more intuitive control over the positions and movements of sound sources, as opposed to the conventional "channel-based" approach.

This is not the initial attempt to spatialize sound in realtime with the Zirkonium. In fact, live-diffusion has been possible and exercised since the first release of Zirkonium in 2006[10]; tentative live diffusion tools, that control the Trajectory editor remotely, were occasionally implemented with Max/MSP, Pd, or SuperCollider for each specific composition. However, the main focus of Zirkonium development always lied on the production of fixed-media composition.

Along with the release of ver. 3.1, we now attempt to provide a universal tool for live-diffusion beyond one-time usage, so that the user can get accustomed to the interface by practicing and working with it on a long-term basis.

Another significant advantage of the iPad is its mobility; the user can freely move around in the performance space. This is especially valuable during sound checks and rehearsals, where the performer can test how specific sound movements are perceived in different parts of the listening space.

2.2.2 Interface

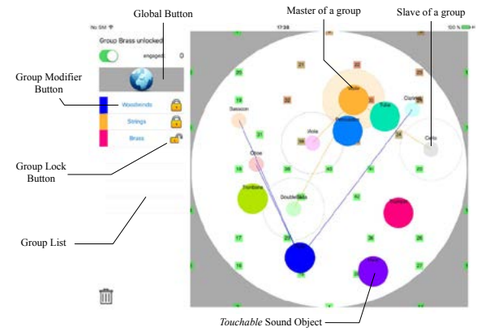


Figure 9. ZirkPad

Figure 9 shows the main interface of ZirkPad. On the right side of the screen, the multi-touch-enabled Dome view is displayed. It synchronizes with the Dome view of the Trajectory editor and visualizes the position of speakers and sound objects as well as the levels of the audio signals that each sound object generates and each speaker receives.

Unlike the Dome view in the Trajectory editor, the sound objects in the Dome view in ZirkPad are touchable; users are able to move single or multiple sound objects by dragging them with the fingers. As soon as ZirkPad recognizes touches on sound objects, it sends OSC messages to the Trajectory editor. In response to the received OSC messages, the Trajectory editor updates the position of the sound objects immediately.

A fundamental problem of live-diffusion with multiple sound objects is that we have got only 10 fingers and we are not able to move them completely independently, although we might want to control the position of more than 10 sound objects simultaneously.

An obvious solution would be to arrange the objects into musically meaningful groups. For example, we could assign 32 sound objects to four groups, each comprising

eight objects. In general, two to four groups seem to be a good quantity for live diffusion, being a good compromise between possible density and complexity on one hand, and clarity and ease of use on the other hand. ZirkPad offers two approaches to move a group of sound objects. The first approach is called *direct group mode*, the second *parametric group mode*.

In the direct group mode the user has to specify one master sound object per group. All remaining sound objects of that group are slave objects. When the master of a group is dragged by a finger, the slave objects follow the master, keeping the same azimuth and elevation offset to the master, in other words, they adopt the rotation mode as described above (2.1.3). The slave objects are displayed smaller in size, so the interface is clearly arranged and not overcrowded. The slave objects are not draggable themselves in order to prevent the user from accidentally moving them. However the user can unlock a group by tapping the respective lock button in the group list. Once a group is unlocked, its master and slave objects can be controlled independently from each other. When the group is locked again, it updates the spatial relationship or the formation of all group members to the current one. In other words, it follows the rule of "free" group mode as described in section 2.1.3. Thus the shape of a group can be changed in an easy way. This gives the user utmost flexibility in handling a group during a performance, while not suppressing the straightforward usage.

In the parametric group mode there is no direct control over the position of a sound object. Instead, the movement of the hand is mapped more indirectly to a movement of the group. With four fingers, the movement of the hand along the Y-axis of the screen controls the span or size of each sound object; with three fingers, the group is rotated; with two fingers, the elevation is changed; with one finger, the group is translated in parallel along the X- and Y-axis.

The parametric group controls can be applied to a single group as well as to several groups simultaneously. There is also a "global" button that activates the parametric controls for all sound objects at once.

It is also possible to establish one to many connections between the Trajectory editor and multiple ZirkPad running on several iPads, so that multiple performers can control a single Trajectory editor simultaneously. This feature is another means to control a large number of sound objects in live situations.

2.2.3 Use Cases

There were already several opportunities where we successfully employed ZirkPad in real-life situations.

The first concert took place on Dec 11th, 2015, in the ZKM_Cube concert hall, where the piece "Next City Sounds Hörstück" was being diffused by ZirkPad. It was a improvisational piece by the IMA, with five stereo channels coming from electronic instruments on stage. We used a single iPad with 10 individual sound objects without group mode.

The second occasion was a workshop at ZKM for high school students. It was especially interesting to see how quickly the students started to feel at ease with the interface and that they were able to use it in a convincing manner, even though those students did not have much experience in musical performances previously.

Another very valuable opportunity was a concert of the Australian collective "Tralala Blip" on Jan 30th, 2016, in the ZKM_Cube. In this concert we successfully utilized ZirkPad with two performers, each controlling one group of sound objects.

3. CONCLUSION AND FUTURE PLANS

For the new release of Zirkonium, we focused on improving usability and live-control capability. We attempted to attain these objectives by revising the software structure, introducing the new GUI components, and implementing a novel iOS App for live spatialization. For the next development phase, enhancements in archive functionality, backward compatibility, and configurability of the hardware settings are planned. In addition, more algorithmic and audio-based approaches for trajectory creation will be further explored. Zirkonium ver. 3.1 public beta is available for free on the website of ZKM | IMA. `http://zkm.de/zirkonium/`.
ZirkPad will be soon available from the App store.

4. REFERENCES

[1] T. Carpentier, M. Noisternig, and O. Warusfel, "Twenty Years of Ircam Spat: Looking Back, Looking Forward," in *Proceedings of ICMC*, North Texas, 2015.

[2] AmbiX. [Online]. Available: http://www.matthiaskronlachner.com/?p=2015

[3] C. Miyama, G. Dipper, and L. Brümmer, "Zirkonium MKIII - a toolkit for spatial composition," *Journal of the Japanese Society for Sonic Arts*, vol. 7, no. 3, pp. 54–59, 2015.

[4] J. C. Schacher, C. Miyama, and T. Lossius, "The SpatDIF library – Concepts and Practical Applications in Audio Software," in *Proceedings of ICMC*, Athens, 2014, pp. 861–868.

[5] M. Puckette, *The Theory and Technique of Electronic Music*. World Scientific, 2007.

[6] C. Ramakrishnan, "Zirkonium: Noninvasive software for sound spatialisation," *Organised Sound*, vol. 14, no. 3, pp. 268–276, 2009.

[7] N. Papters, J. Schacher, and T. Lossius. (2012) SpatDIF specirfication V0.3. [Online]. Available: http://spatdif.org/specifications.html

[8] F. Bayle, "À propos de l'Acousmonium, Recherche Musicale au GRM," in *La Revue Musicale*, vol. 394-397, 1986, pp. 144–146.

[9] S. Wilson and J. Harrison, "Rethinking the BEAST: Recent developments in multichannel composition at Birmingham ElectroAcoustic Sound Theatre," *Organised Sound*, vol. 15, no. 3, pp. 239–250, October 2010.

[10] C. Ramakrishnan, J. Goßmann, and L. Brümmer, "The ZKM Klangdom," in *Proceedings of NIME*, Paris, 2006, pp. 140–143.

A 3D Future for Loudspeaker Orchestras Emulated in Higher-Order Ambisonics

Natasha Barrett
Department of Musicology, University of Oslo.
nlb@natashabarrett.org

ABSTRACT

The majority of acousmatic music exists in stereo: a format best performed over a loudspeaker orchestra. Although multi-channel formats are popular, many composers still choose stereo and link their work to diffusion performance practice. Yet a growing trend in loudspeaker systems is the permanent, high-density loudspeaker array (P-HDLA). Constructed from similar speakers, evenly distributed around the space, P-HDLAs maximize audience area and eliminate time-consuming setup time. Although compatible with most multi-channel formats, they pose a practical and musical dilemma for stereo sound diffusion.

Looking to the past and future of stereo acousmatic composition and performance, I have designed the 'Virtualmonium'. The Virtualmonium emulates the principles of the classical loudspeaker orchestra in higher-order ambisonics. Beyond serving as an instrument for sound diffusion, composers and performers can create custom orchestra emulations, rehearse and refine spatialisation performance off-site, and discover new practices coupling composition with performativity. This paper describes the Virtualmonium, tests its practical application in two prototypes, and discusses the challenges ahead.

1. INTRODUCTION

Historically, the loudspeaker orchestra has served as an instrument for performers to project and enhance spatial contrast, movement and musical articulations latent in their acousmatic compositions. To achieve these results, the orchestra combines diverse loudspeakers placed at different distances and angles from the audience area. The performer draws on the combination of spectral and spatial changes in the music, loudspeaker characteristics, room acoustics, and how changes in the precedence effect and directional volume influence listeners' perception of the spatial scene. There are few large loudspeaker orchestras. Some examples include the GRM Acousmonium and the Motus Acousmonium in France, BEAST in the UK, and the Musiques & Recherches Acousmonium in Belgium.

Fixed-installation, high-density loudspeaker arrays (P-HDLAs), constructed from similar speakers evenly distributed around the space, are becoming more common. P-HDLAs are diverse in their design: ZKM's 43-speaker Klangdom Hemisphere at the Center for Art and Media in Karlsruhe, Germany, can be set in contrast to IRCAM's 75-speaker 3-D cuboid in the Espace de Projection, Paris, France. Higher-order ambisonics [1] (HOA) can be used to project spatial information over such systems, where increasing numbers of directional components in the spatial encoding facilitates an increasingly accurate representation of the spatial scene over a listening area suitable for public audiences. Furthermore, recent decoder developments [2] allow loudspeaker layouts to depart from the geometric restrictions demanded by conventional decoding formulas, accommodating most P-HDLA layouts. Near-Field Compensated HOA (NFC-HOA) which, when implemented in higher orders of ambisonics, can focus sounds in front of the loudspeaker array and improve image stability for sounds outside the array, can be useful. Although implementing mathematical solutions in real-time presents practical challenges [3], developments in the technology continue to advance. Wave Field Synthesis has been shown to more clearly stabilise the image than HOA [4], yet there are practical considerations concerning spectral bandwidth, verticality, and loudspeaker numbers.

The Virtualmonium emulates the principles of the acousmonium in HOA. Although it is unrealistic to emulate every detail of a real acousmonium, the goal is to offer the same affordances in terms of performance and spatial results. Role models are drawn from two categories of acousmonium design: one where loudspeakers form a frontal stage with fewer speakers surrounding the listeners, such as in the Gmebaphone [5]; the other where loudspeakers are more evenly distributed throughout the space, such as in the BEAST system [6]. Despite their differences, these systems feature common practical and performance goals:

- To combine loudspeakers of different power and frequency response, and to utilise the geometry and acoustics of the concert space. Over a correctly designed setup, speaker characteristics and acoustic features will highlight spectral characteristics in the music, which will then tend to 'spatialise itself'.
- To facilitate interesting and stable, although not identical, phantom images across a broad listening area.
- To facilitate the performance of musical features such as distance, motion, envelopment and elevation, to unfold the layers of a stereo mix-down in space, and to articulate changes in musical structure.
- To focus on performance, where the resulting complexity can only be controlled through the act of performing-listening.

2. EMULATION

To emulate the loudspeaker orchestra we can address four areas: (1) loudspeaker locations as azimuth and distance from the listener, (2) loudspeaker rotation and its effect on directional volume, frequency and room stimulation, (3) a room model to aid the impression of distance and for speakers to stimulate diffuse wall reflections, and (4) loudspeaker colour, power and radiation pattern. These four categories can then be grouped into loudspeaker emulation and spatial emulation.

2.1 Processing overview and performance control

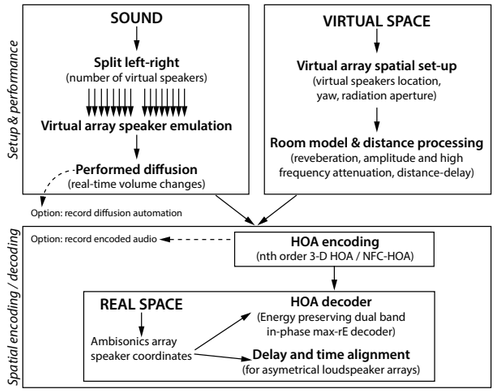


Figure 1. The Virtualmonium processing overview showing loudspeaker emulation, performance input, virtual room model, spatial encoding and spatial decoding

An overview the Virtualmonium is illustrated in figure 1. In diffusion, a mixer is the traditional instrument, where the stereo source is split over many individual channels, and where one fader controls the volume of one loudspeaker. The Virtualmonium is controlled using a set of MIDI faders, emulating the function of a traditional mixer. Expansions to mapping one fader to one speaker have been described in the Resound project [7] and in BeastMulch [8]. These approaches can be easily layered into the Virtualmonium interface.

2.2 Spatial emulation

Spatial emulation is carried out using existing tools running in MaxMSP, including IRCAM's Spat package [9]. Spat's sources are used to position and rotate the virtual loudspeakers in space, define a radiation aperture, and to stimulate a room model. A real-time ambisonics room model is beyond the scope of our current computer resources. Instead, IRCAM's reverberation model combining convolution and panned early reflections is used. [10] For each virtual loudspeaker, distance-derived delay, amplitude attenuation and air absorption, and the appropriate direct and reverberant signal levels are calculated from loudspeaker location, radiation and aperture.

Although numerical methods may predict source localization for different orders of ambisonics, few studies have tested subjective evaluations. Frank et al. [11] found similarities between a 5th order system and a real sound field when centrally located listeners were asked to localise sounds in their frontal plane. Wierstorf [4] showed greater variation for listeners located in an area spanning from the centre to the perimeter of the loudspeaker array. With these results in mind, the Virtualmonium will require at least a 5th order loudspeaker array. The current prototype applies 6th order 3-D ambisonics.

2.3 Loudspeaker emulation

Although the loudspeakers of the Ambisonics array will colour the sound, we assume that they are neutral.[1] Gathering information about the loudspeakers, and their driving amplifiers that we want to emulate, is not trivial. Toole [12] lists 11 descriptive criteria, amongst which we find frequency-amplitude responses, non-linear distortions, and power handling. For the first prototype, a straightforward approach emulated speaker properties by convolving sources with impulse responses (IRs) as well as by applying filters estimated from frequency plots. We have not begun our own loudspeaker measurements and for proof of concept, IRs from the online database at the 3-D Audio and Applied Acoustics Lab at Princeton were used [13]. In the database, measurements were made for each 5-degrees in the horizontal plane. In the first prototype, to understand how the virtual-speaker orientation interacted with the total spatial scene, rather than 72 virtual sources of a 5-degree aperture, single IRs from 0 degree rotation (or filters) were used.

Although the aim is to approximate rather than reproduce a real loudspeaker in ambisonics, close replication is desirable in terms of history and tradition, both for rehearsing on virtual arrays, and aligning with the expectations of the performer. Cross-comparisons between the emulation speaker and its real-world counterpart are underway, and with this in mind IRs from three speaker models that were also readily available were chosen: KEF LS50, Genelec 8030A and JBL Professional PRX635.

Compression and limiting were used to approximate dynamic performance. However, without visual cues informing users of the dimensions of the virtual speaker, this processing tended to confuse listeners unfamiliar with the system, and was removed from the first prototype. So far, neither frequency dependent phase nor volume dependent harmonic distortion have been addressed.

3. TESTING THE VIRTUALMONIUM

Prototype-I has been tested in performer and listener studies. Tests were carried out in the 3-D lab at the Department for Musicology at the University of Oslo: a room 8m x 5m x 3m housing a 47-speaker Genelec 8020 array. Besides heavy drapes, the room is acoustically untreated, but with a small audience the reflections are significantly reduced. Despite the suboptimal room geometry, a 6th order 3-D HOA decoding functions well. In this space it is not possible to setup a real acousmonium for direct comparison to the 3-D HOA array, so tests were designed to address the affordances of the system.

[1] In new work, compensation filtering is being tested.

3.1 Listener and performer test method

Tests were carried out over a small Virtualmonium, consisting of 15 virtual speakers as shown in figure 2. The loudspeakers are numbered as they appeared on the faders and their radiation pattern shown by the white shading. Speakers 13 and 14 are elevated, emulating tweeter grids. One MIDI fader controlled each virtual speaker.

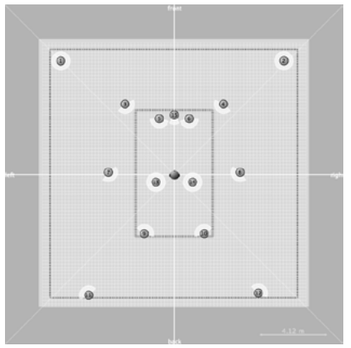


Figure 2. The virtual loudspeaker test array: the inner rectangle is the boundary of the real Ambisonics array and the outer rectangle is the room model (20m x20m x10m).

Unlike many other countries, Norway lacks a diffusion performance tradition. Rather than advertise for volunteers without performance experience, six composers with diffusion experience were invited to participate. Performers were tested on their ability to execute a number of spatialisation techniques idiomatic of classical diffusion practice. A group of listeners, and the performers themselves (as self-evaluation), rated the results.

Appropriate custom-made sounds were allocated to each spatialisation technique.[2] Table 1 provides an overview. Performers were centrally located and allowed to practice each task prior to evaluation. Listeners were seated, spanning a diameter of 2/5 the width of the loudspeaker array. Before each evaluation, an omnidirectional version of each sound was played as a reference. To be consistent with the realities of concert diffusion, performers were allowed to interpret the spatial performance actions as expressions of the composed sounds. Some sounds were specified as statically located, others as changing from one spatial state to another. For example, an interpretation of 'front narrow then widening' may widen with a front, side, rear or surround bias. Some of the performance actions involve the term 'gesture'. This connects to acousmatic composition and performance practice, where 'gestural' sounds, which contain dynamic changes in spectromorphology, should be appropriately interpreted in the spatial and volume dynamics of the performed motion. However, performers were made aware that the speed of the action was determined by the duration of each test sound. If, for example, 'Gesture, medium speed' were to be performed too slowly, the sound would have ended before it had moved much in space, resulting in a low score.

[2] http://www.notam02.no/natashab/VM_tests1.zip

Spatial performance action	Sound type
Front close	Short impulse with repetition v1
Sides close	Short impulse with repetition v1
Front narrow then widening	Short impulse with repetition v2
Rear narrow then widening	Short impulse with repetition v2
Wide space	Short impulse with repetition v3
Move mid to distant space	Gesture move mid to distant
Gesture, medium speed	Gestural archetype, medium
Gesture, fast speed	Gestural archetype, fast
Gesture, curved (circular)	Gestural archetype, long
Spatial perspective shift	Perspective shift
Immersion wash	Full bodied wash
Immersion detailed	Detailed textural flow
Intimate sensation	Whisper close
Erratic motion	Erratic texture
Layered space, simple	Two-part complex environment
Layered space, complex	Multi-part complex environment

Table 1. Spatial performance tests and sound types

The evaluation was made on a scale from 0 to 4, where 0 reflected no difference to the reference, 1 was a poor interpretation and 4 reflected a performance matching the specified action. As both the performance and evaluation involved subjective components of interpretation and quality, subjects were advised that a score of 2 would indicate similarities to the action, but that the results were vague.

3.2 Results from performance and listener tests

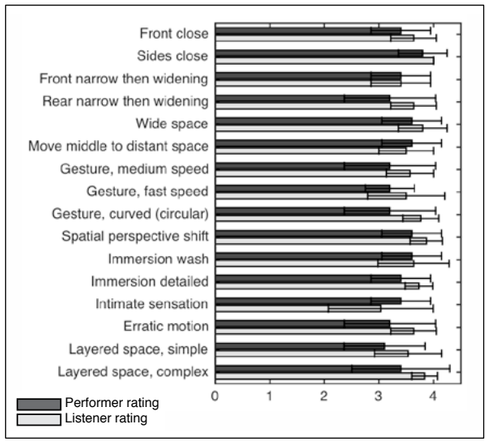


Figure 3. Test results for performers and listeners

Figure 3 shows the mean and standard deviation for all evaluations of the 16 tests. Although only six performers were involved, we can see that all performance actions were relatively clearly articulated, that performers generally scored themselves harder than listeners, and that moving gestures scored slightly lower than static sounds. The lowest listener score was for intimate sounds. Layered spaces, which can be tricky to project over a real loudspeaker array, scored well.

3.3 Concerts and audience responses

Concerts present a more complex set of challenges for both performer and listener. Beyond expressing musical

spatial gestures, performers and audiences may become acquainted with the colour and diversity of the loudspeakers as an orchestral ensemble, where the configuration evokes anticipation and expectation in both performance and listening process.

For the concerts, the lab doubled as the venue. Due to its size, the audience was limited to 14 (11 central and three peripheral seats). Works drew on standard repertoire familiar to the performer from real acousmonium experiences (such as works by François Bayle and Horacio Vaggione). Concerts were spatialised over setups of between 20 to 30 speakers, custom designed for each concert. The diffusion was perfected in advance with the performer centrally located, and the results encoded to a sound file. In the concert, the performance position was removed for the audience to occupy the optimal listening area, and the work decoded to recreate the performance.

The audience spanned a range of backgrounds and age groups. Their experiences were investigated through post-concert conversations. Listeners unfamiliar with the diffusion genre assumed that the stereo compositions were native 3-D, while familiarised listeners were positive to the spatial-musical projection. Experienced listeners were generally satisfied, although distance, reverberation and colouration arose as topics of discussion.

3.4 Prototype-II

Based on the results from the first prototype, Prototype-II implemented a number of spatial and speaker developments, which have been tested informally and in the most recent concerts.

3.4.1 Loudspeaker directional frequency response

Directional-frequency is an interesting feature as it interacts with the virtual room as well as changing the colour based on speaker-rotation. To approximate to a directional frequency pattern, each virtual loudspeaker is represented by seven directional IRs of 60-degree rotation, where each filters a copy of the source. The outputs are first treated as omnidirectional, then narrowed by a radiation aperture and rotated corresponding to the relative direction of the original IR.[3]

3.4.2 Proximity

NFC-HOA was introduced to project virtual sources a small distance in front of the real loudspeakers. The sensation of proximity is however not the same as for real near-field speakers. The technique also introduced some perceptual complications: although sources positioned outside the ambisonics loudspeaker array served a large listening area, those focused inside the array appeared to compromise this area. For these sources, when listening from a location satisfactory for normal HOA, for NFC-HOA the source slightly overshadowed other information in the space.

[3] In theory, the aperture of each source would then be set to 60 degrees. However, Spat's directivity-derived pre-equalization, which is part of the aperture processing, 'leaks' beyond the specified range. In our emulation this serves as a transition zone between directional IRs and it was found that smaller apertures were useful, set by ear.

When encoding NFC-HOA the radius of the loudspeaker array must be specified at the outset, which complicates the transfer of encoded spatialisation performances between HDLAs. Although the Spat package includes a radius compensation adaptor (tested informally to function with a tolerance of between 50%-200% of the encoded array radius), it is spatially more accurate to record performed fader automation, and in the concert, couple encoding with decoding in real-time. This has however so far lead to temporal inaccuracies due to automation densities being insufficient to capture spatial changes with millisecond precision.

As our strongest judgement of distance is on a relative rather than absolute scale, we can be less concerned with absolute proximity and turn our attention to contrasts. In prototype-II signals are directly routed to a selection of real loudspeakers. These add 'sharper' points to the palette of spatial contrast, but require strategic placement to avoid appearing overtly conspicuous.

3.4.3 Distance

The Virtualmonium relies on a room model to project distance and to emulate rotational and directional characteristics of loudspeaker constellations. HOA removed the boundary delimited by the real loudspeakers and real room remarkably well, allowing the space of the music to easily extend beyond the confines of the listening space.

When specifying the room model, two decisions need to be made. The first concerns the room model in relation to the real room acoustics, where in larger spaces the two may conflict. In such circumstances it may be interesting to calibrate a room model such that it appears to 'extend' the real room acoustics. In very large spaces, a room model could mimic the real space, only to be used when stimulated by non-direct virtual loudspeakers. These implementations will be tested when we have access to a larger concert space.

The second consideration concerns distance-related delay, where the signal from virtual sources further away should be correctly delayed to maintain the ratio between distance, sound-pressure level of the direct sound and the reverberant sound. In prototype-I a delay-related colouration was sometimes audible when two virtual loudspeakers of different distances played simultaneously. In a real acousmonium, the complexity of room acoustics and loudspeaker characteristics eliminates this effect. In prototype-II, virtual-speaker directional filtering improved the results. Future improvements will address frequency-dependent phase and volume-dependent harmonic distortion in speaker emulation, as well as adjustments to the room model.

3.4.4 Other spatial formats

Many composers continue to work in a fixed multichannel format. Some composers have learnt to adapt their multichannel works to each performance situation. Other composers require a precision setup that is often unrealistic. To address these matters the Virtualmonium has been tested for multichannel emulation. The works performed have included Arne Nordhiem's 4-channel 'Solitaire' (1968) and Tor Halmrast's 12-channel 'Aqueduct' (1992). 'Solitaire', originally made for a large installation space, is a challenge to project adequately over four loudspeakers in a smaller space. In the Virtualmonium, the four channels were positioned inside an HOA room model approximating to the acoustics of the original venue. The quadraphonic source no longer sounded as four points but rather created a clearly spatialised sound field. Halmrast's 'Aqueduct', composed for the Norwegian Pavilion at Expo 92 Seville, Spain, consists of an eight-channel cubic loudspeaker setup with four high frequency overhead channels. The eight channels were positioned in their original geometry as ambisonics sources. For the overhead channels, to enhance directionality and proximity contrast, we found that direct speaker feeds rather than virtual ambisonics sources were most effective, taking advantage of the normally troublesome precedence effect.

4. DISCUSSION AND FURTHER WORK

4.1 The visual paradox

Although acousmatic concerts may be held in near darkness, lighting is often used for scenographic effect. For both performer and listener, visual information influences our spatial hearing in terms of the assumed direction and distance of the loudspeaker. Yet the Virtualmonium creates invisible virtual loudspeakers within an Ambisonics sound field, where the listener paradoxically sees the Ambisonics array while the sound appears from a zone of 'empty space'. This conflict between auditory and visual cues is apparent in our lab. Furthermore, the visual loudspeakers in a real acousmonium serve as landmarks that systematise the performance and assist our memory of how movements of the faders translate to changes in the spatial image. The loudspeaker as an object also reminds us of its non-linear response. As a performance aid, a 2-D image of the virtual loudspeaker was tested, but this tended to reduce 'performance through listening'.

4.2 Size of HDLA and size of Virtualmonium

Our ability to perceptually differentiate virtual sources draws on angle, distance and colouration. Although we can estimate the number of virtual loudspeakers each system can support based on angular differences by calculating the angular blur for each order of Ambisonics, the effect of distance and frequency can only be investigated in practical tests. To date, work has been conducted over a 3-D array. In future work, a comparative study will explore the implications of 2-D decoding.

4.3 Composition

As a composer, despite working extensively with ambisonics, my compositional language has been invaluably enriched by experiences of stereo sound diffusion performance: reconciling spatial concept with spatial reality, an awareness of sound in space, and the journey of the composition from the seeds of its creation to the concert-goer and a multitude of sound design considerations. Although stereo and 3-D composition have much in common, each format carries unique qualities concerning the way in which a composer approaches spatial objects and scenes, aesthetical and practical intentions, and in the approach to audiences and concerts. In new work, the Virtualmonium will be explored as a compositional tool for the 3-D spatial expression of stereo sources alongside native 3-D ambisonics materials.

5. REFERENCES

[1] Daniel, J., and S. Moreau. 2004. "Further Study of Sound Field Coding with Higher-Order Ambisonics." *116th Convention of the Audio Engineering Society, 8–11 May 2004, Berlin*.

[2] Zotter, F., Pomberger, H., Noisternig, M. "Energy-Preserving Ambisonic Decoding". *Acta Acustica United with Acustica*. 98:37–47. 2012.

[3] Favrot, S., Buchholz, J. "Reproduction of Nearby Sound Sources Using Higher-Order Ambisonics with Practical Loudspeaker Arrays." *Acta Acustica United with Acustica*. 98:48–60. 2012.

[4] Wierstorf, H., Raake, A. and Spors, S. "Localization in Wave Field Synthesis and higher order Ambisonics at different positions within the listening area." *Proc. DAGA*. 2013.

[5] Clozier, C. "The Gmebaphone Concept and the Cybernephone Instrument". *Computer Music Journal*, 25:4, pp. 81–90. 2001.

[6] Harrison, J. "Diffusion: theories and practices, with particular reference to the BEAST system". *eContact!* 2.4. 1999.

[7] Mooney, JR and Moore, D. "Resound: open-source live sound spatialisation". *Proc of the ICMC. Belfast, UK*. 2008.

[8] Wilson, S., and Harrison, J. "Rethinking the BEAST: Recent developments in multichannel composition at Birmingham ElectroAcoustic Sound Theatre". *Organised Sound* 15, pp 239-250. 2010

[9] IRCAM. 2015. http://forumnet.ircam.fr/product/spat/ Accessed Aug 2015.

[10] Carpentier, T., Noisternig, M., Warusfel, O. "Hybrid reverberation processor with perceptual control". *Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx-14)*, 2014. Erlangen, Germany.

[11] Frank, M., Zotter, F. and Sontacchi, A. "Localization Experiments Using Different 2D Ambisonics Decoders". *25th Tonmeistertagung - VDT International Convention*. 2008.

[12] Toole, F. "Loudspeaker Measurements and Their Relationship to Listener Preferences". *AES* Vol. 34. No 4. 1986.

[13] Tylka, J., Sridhar, R., Choueiri, E. "A Database of Loudspeaker Polar Radiation Measurements". *AES Convention 139*. 2015.

[14] Johnson, B. and Kapur, A. "Multi-Touch Interfaces For Phantom Source Positioning In Live Sound Diffusion." *Proc. of NIME, Kaaist, South Korea*. 2013.

Extending the Piano through Spatial Transformation of Motion Capture Data

Martin Ritter
University of Calgary
Calgary, Alberta, Canada
martin.ritter@ucalgary.ca

Alyssa Aska
University of Calgary
Calgary, Alberta, Canada
alyssa.aska@ucalgary.ca

ABSTRACT

This paper explores the use of motion capture data to provide intuitive input to a spatialization system that extends the resonating capabilities of the piano. A camera is placed inside of an amplified piano, which tracks the motion of internal mechanisms such as hammer movement. This data is processed so that singular pitches, chords, and other motions inside the piano can produce sonic results. The data is subsequently sent to an ambisonics software module, which spatializes the piano sounds in an effort to provide a meaningful connection between the sound location and the perceived vertical frequency space produced when creating sound on the piano. This provides an audible link between the pitches performed and their location, and an effective option for integrating spatialization of the piano during performance.

1. INTRODUCTION

Motion tracking systems have been explored extensively by the New Interfaces for Musical Expression (NIME) community, and inputs for these systems have included devices such as cameras [1], Leap Motion [2, 3], and Xbox Kinect [4,5]. The intent of such systems has generally been to provide an engaging musical interface for a performer to interact with, usually as an analogue for an acoustic instrument, or to serve other interactive purposes. The software discussed in this paper extracts and sonifies motion capture data from the piano, providing performers and composers the ability to extend the spatial and resonant characteristics of the instrument by tracking the its mechanical actions, which occur as a result of the performer's gestures. This data is obtained by placing a camera on the inside of an amplified piano to capture internal hammer and damper movement, and extended techniques that require performers to play inside of the piano.

This use of motion capture differs from other means of piano performance tracking employed by the authors thus far, which have included following the hands during performance with a camera [6], and using the Leap motion peripheral to track finer grained motions of the hands and fingers [2,3]. These precedents were primarily designed to create meaningful systems of interaction between the performer and the electronics. Tracking the inside of the piano captures the motions of the instrument, rather than the performer. This provides a different approach and data set; while the hammers will be activated on a similar horizontal axis as the hands that depress the keys, the motion is less susceptible to data fluctuation, as it is a rigid mechanical system rather than human action that is being tracked. This increases the potential for accurate tracking of singular pitches or pitch areas. Additionally, the mechanics of the piano are motionless when they are not in use, whereas humans are more likely to generate subtle motions during performance, even when they are directed to remain motionless. Therefore, tracking inside the piano increases data accuracy while decreasing jitter since it prevents the capture of these extra-musical gestures. Finally, since the strings and sounding board of the piano have been used to create resonance effects in several contemporary creative compositions, the choice to track motion from within the piano was also an aesthetic one, expanding upon these ideas of piano as a spatial instrument [7, 8]. It should be noted that this software focuses on the spatial transformation of live, acoustic pianos during performance. While it may seem more practical and accurate to track the depression of keys on a MIDI piano, this software aims to extend the capabilities of the acoustic piano. The module was designed originally for a specific piano trio that included electronics that aimed to create a link between the spatilization and the musical gestures while avoiding the extra-musical motions. A Disklavier would also be a presumably viable option for tracking via MIDI, but not every concert hall and venue has this option available. Since future explorations with this module will include the performance and spatialization of standard concert repertoire as well, this software must work with an acoustic concert piano.

2. PRECEDENTS - IMUSE

The Integrated Multimodal Score-following Environment (IMuSE) was a SSHRC-funded project under the supervision by Drs. Hamel and Pritchard at the University of British Columbia, Vancouver, Canada [6, 9]. The system was primarily designed to aid in the rehearsal and performance of score-based interactive computer music compositions. IMuSE incorporates several different software components such as NoteAbilityPro [10, 11] for the notated score (both the traditional score for the performer as well as the score for the electronics) and Max/MSP [12] or pd [13] for the performance of the computer generated sounds as well as the analysis, matching, and networking of the tracking data to NoteAbilityPro.

While this project was conceived as a score-following tool, it quickly became clear that it could be used for creative purposes as well. Various pieces were written using the capabilities of tracking the performer. [14], [15], [1], and [16] use these tracking technologies, which were developed specifically for score following; the are instead employed to create different musical effects. [14], [15], [16] use the tracked motion to create data to – among other things – synthesize a second piano to various degrees while [1] used the hand gestures as a compositional concept, which linked the motion tracking data to many electronic processes, so that the electronics were organically linked to the music. While these pieces were succesful, one constant issue each composer had to deal with was auxiliary movements by the performer, which could interfere with the tracking if they were picked up by the system (e.g. the performers' heads had the tendency to enter the tracking area).

3. CAPTURING HAMMER MOVEMENT DATA

IMuSE was concerned with tracking the approximate location of the pianists hands for the purpose of score following. This software, in contrast, indirectly tracks the motion of the pianist by concentrating on the mechanical movement of the hammers. A camera is placed inside of the piano, capturing the entire row of hammers.

The current version of the software allows the user to crop and rotate the incoming video stream so that only the hammers are visible. As a side effect, computational time is greatly reduced by only analyzing subregions of the entire video frame. Rotation may be necessary depending on the placement of the camera. The cropped image is then analyzed using very simple but effective techniques. First the video stream has to be prepared for analysis. The absolute difference of a grayscale version of the picture is computed, which means that only pixels that are in motion from one frame to the next are visible and used for further analysis. Next, the image is binarized using thresholding and smoothing, which removes most of the unwanted noise in the video signal. In an effort to exaggerate the motion, a morphological close operation may be added to the video, which has also the side effect of minimizing the remaining noisy components. At this point the actual analysis takes place. The cropped frame is divided into customizable regions (by default this is one region per hammer), each of which is evaluated for the amount of active pixels. If the amount of active pixels exceeds a customizable threshold, the region is marked as on for the current frame. A list of such regions is compiled and output for use in the sonification system for each frame of the video.

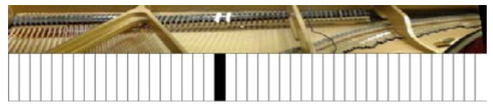


Figure 1: Top: Cropped and rotated image with superimposed movement data (white); Bottom: tracking region divided into 44 discrete areas with currently tracked region in black.

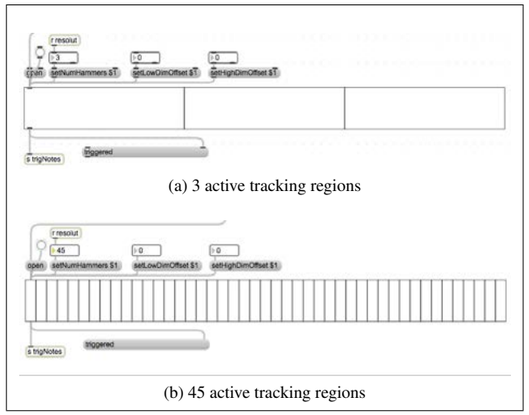
4. SPATIAL TRANSFORMATIONS MADE USING DERIVED DATA

All of the data produced by the camera and analyzed by the computer vision software is subsequently sent to a spatilization module developed by the author in Max entitled AAAmbi, which uses the ambisonics tools for developed at ICST in Zürich [17]. AAAmbi provides a versatile and accessible interface in which users can send specified spatialization data through an ambisonics encoder and decoder. These messages consist of information that modifies parameters such as the azimuth angle, height, and distance of sources, as well as more complex options such as grouping of sources and spatial trajectories. AAAmbiPianoHammers is a module that works in conjunction with AAAmbi, and they are designed to send and receive messages from one another.

4.1 Inputs to Spatialization

The motion tracking data that is used for AAAmbiPianoHammers includes hammer movement information as well as the size of motion. Whole number integers sent as lists make up the hammer tracking data, and floating point numbers are sent that represent the centre of the location of active pixels. The total number of active pixels is also sent. The hammer action integers are filtered using this number of active pixels, and only movements that contain less than 250 active pixels are sent to the hammer-tracking algorithm. This enables precise location detection so that the hammer tracking can correspond to approximate pitch. The integers representing the hammer movement are sent as lists whose lengths vary, depending on the number of keys depressed. For example, if three hammers or regions are reported as being on or active, the software will report a list of three different values. The hammer movement is sent through an algorithm that segments the data into several regions, which can be specified by the user. This affects the grain of the spatialization because larger segmentation regions yield a much higher resolution and therefore more pitch detail. Lower numbers will create a more general link between active areas of the piano and localization. At the outset it seems that one would always want as high resolution as possible, however, one potential drawback to higher resolution is that the likelihood of false positives is increased. Therefore, the user should balance their resolution and accuracy needs to determine an appropriate number of segments.

In addition to user flexibility regarding resolution, the software also allows specification between fixed and relative spatialization. In fixed spatialization mode, the middle of the keyboard always corresponds to the middle of the sound field and pitches are generally placed in the sound field outwards from the centre as they become higher and lower. Relative, or moveable, spatialization enables the pitch region associated with the centre of the sound field to vary depending on which pitches are active at any given time. For example, in a relative system, if the performer plays a series of pitches beginning on the low end of the piano, the lowest pitch will be initialized as the centre of the sound field, and all pitches above it will be treated as an increases and decreases in values.



(a) 3 active tracking regions

(b) 45 active tracking regions

Figure 2: Customizable tracking regions. (a) = 3 active tracking regions; (b) = 45 active tracking regions;

4.2 Effect of data on spatial parameters

AAAmbiPianoHammers has two audio outputs, but there are eight points in AAAmbi that must be spatialized. Therefore, the user should make sure that there are eight inputs available in their AAAmbi module. The inputs to AAAmbi default to one through eight, but the user can specify these to be any inputs as needed. The hammer number integers are grouped within AAAmbiPianoHammers. The length of the group, indicating the number of active hammers, determines the volume of the piano amplification. The mean of this group is used to determine the azimuth angle. The lowest and the highest of the triggered hammer numbers is calculated, and these numbers are used to determine Cartesian coordinates. A midpoint of the user-defined tracking resolution (number of hammers) is calculated, and any numbers below that midpoint are scaled in such a way that the lower the number, the closer it will be towards the bottom left corner on a cartesian plane, which results in a very distant, left, and rear sound. For numbers that are higher than the midpoint, the higher the value, the closer to the top, right area of the cartesian plane the sound will be placed. These sounds are then spread around the cartesian plane relative to the highest and lowest values. A larger spread between the higher and lower value will result in more perceived distance between the sounds (and distance between sounds on the cartesian plane).

This will also result in more immersive sound, because it changes the perceived size of the sounding objects. When a small cluster of notes, or a singular note is struck, this will create a small value between the lowest and highest notes, increasing the distance and giving the impression of a smaller object. Therefore, this method of spatilizing the hammer data presents a viable way to provide localization of sounds that is closely correlated to the material performed.

4.3 Extended techniques and special effect

The motion tracking software also tracks the location and degree of movement and made by the performer as an alternate method to the hammer movement, and one that is most effectively used for larger gestures. AAAmbiPiano allows for both of these parameters to affect spatial transformation as well. The degree of movement, determined by the number of active pixels, affects the distance parameter in the AAAmbi module when it is consistently greater than 250 (in effect, when the hammer regions are not on for tracking). This has the effect of closer proximity of the sound as linked to more motion, and decreased presence for less motion. This feature is accessible by performing very broad, uniform gestures, such as extended techniques inside the piano. Plucking a string, for example, results in approximately 300-500 active pixels, which is significantly higher than the 75-250 active pixel averages for a singular depressed note.

Because the hammer tracking filters out motion that remains consistently over 250 active pixels, a gross motor action such as plucking the string string provides too many active pixels to be for the data to be spatialized using the normal hammer tracking algorithm. Therefore, larger motions use the number of active pixels to determine the distance of the sound from the centre, and the location of movement along the horizontal axis to determine the localization of the sound. Plucking a string near the high end of the keyboard and then at the low end would therefore result in the following approximate sonic trajectory:

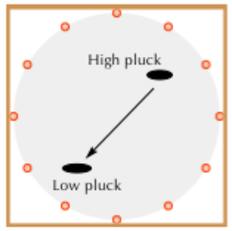


Figure 3: Sample of spatial trajectories of extended techniques performed inside of piano.

Tracking these wider motions is less precise than tracking the hammers, and since it does involve human action there is a wider amount of variability from action to action. Each performer will perform the task slightly differently. Therefore, general rather than specific localization algorithms are actually more effective because they are more predictable.

4.4 Reverberation and pedal trigger

Using the pedal also provides a very high number of active pixels, generally in a range greater than 1000. The use of the pedal can have two results: 1) activating a message, pedalTrigger, which the user can then use for any purpose, and 2) activating the default, predefined action of affecting the wet/dry of reverberation. The pedal trigger activates only when the pedal is raised and then lowered within 1300ms. If the pedal is held for longer than 1300ms, reverberation saturation is initiated, which raises the wet balance and lowers the dry balance. This reverb effect was selected because of the natural effect of the sustain pedal, which is to increase the resonance of the piano and in effect, the length of the notes. Reverb serves a very similar purpose.

5. APPLICATIONS

The primary uses of this software include creative projects, such as musical compositions involving electronics, installations, or sound art pieces using the piano. It is intended for use in any live performance or live installation where a piano is used. This would not be classified as a hyperinstrument (such as those designed at MIT by Tod Machover), because the system is more of an extension of the resonant and spatial capabilities of the piano rather than the performance capabilities [18]. The piano hammer tracking would be effectively implemented in compositions for live instruments and electronic sound and installations that explore space and include user input. The system does have consistent links between the data and the spatial trajectory of the sound, but the customization options allow for variability if the user wishes to obtain different results and more dynamism of the spatial transformations. Additionally, further customization is in development, which will allow composers and developers more freedom. Performers and improvisers can also use the system as a means of extending their instrument, and the system could be used during performance of the standard repertoire with enhanced spatial components. This software has currently been used in a piano trio by the author and will be involved in a major poly-work under development for voice, flute, cello, and piano.

6. FUTURE DEVELOPMENTS

Expansions upon these modules are in development, including more user customizability to allow for dynamic inputs, more spatialization options, and the integration of sonic effects within the module. Refinements of the motion tracking are always ongoing, especially as more works are composed and performed using the software. Another development involves the inclusion of dynamism, which would allow for parameters of the software to be modified by the user in real-time. Future developments of the motion capture include the isolation and filtering of the hammer-off motions, which would make the hammer tracking data more precise and prevent duplicate data.

7. CONCLUSIONS

Motion capture interfaces thus far have mostly been used creatively for the purposes of gestural control and meaningful human-user interface solutions to electronic instruments. The system described in this paper makes use of mechanical tracking that occurs as a result of performance, but is not tracking the performer directly. This provides a different solution to performance gestures and a different use of gestural data that enables spatialization of the motion of the sounding body itself, rather than the enacting body. The two are coupled in the case of the piano, as the depression of a key by a performer is connected to the hammer, which strikes a string to produce a musical sound. The piano body then resonates the sound. Rolf Bader described, in his article Synchronisation and Self-Organization, that frequencies of musical instruments are either determined by a generator (the energy that generates the sound) or the resonator (the body that sustains the sound) [19]. An instrument such as the saxophone, for example, requires energy from breath (the generator) and is sustained by the size of the tube (the resonator), which is determined by depressing keys. The resonator therefore determines the pitch of the saxophone. A violin, in contrast, has as its pitch determinant the length of the string, which is also the energy producing body, or generator. However, Bader further discusses that systems can slave one another, preventing either a generator or resonator from being the sole determinant of pitch, and allowing them to couple. The piano is somewhat more complicated when abstracting this principle to motion tracking; the hand action and the hammer action (which is the actual mechanism responsible for the sound) are geographically and visibly separated by a physical barrier (the piano body), therefore, they are de-coupled from a gestural perspective. This makes the tracking of the hammers to serve as mapping data an option in which sounds correspond mostly, but not entirely to the visual stimulus.

8. REFERENCES

[1] A. Aska, “Concurrent Shifting,” Calgary, 2012, composition for piano and motion tracking.

[2] M. Ritter and A. Aska, “Leap Motion As Expressive Gestural Interface,” in *Proceedings of the International Computer Music Conference*, Athens, 2014.

[3] ——, “Performance as Research Method: Effects of Creative Use on Development of Gestural Control Interfaces,” in *Proceedings of the Practice-Based Workshop at NIME’14*, London, 2014.

[4] M.-J. Yoo, J.-W. Beak, and I.-K. Lee, “Creating Musical Expression using Kinect,” in *Proceedings of New Interfaces for Musical Expression*, Oslo, 2011, pp. 324–325.

[5] A. Hadjakos, “Pianist motion capture with the Kinect depth camera,” in *Proceedings of the International Conference on Sound and Music Computing*, Copenhagen, 2012.

[6] M. Ritter, K. Hamel, and B. Pritchard, “Integrated Multimodal Score-following Environment,” in *Proceedings of the International Computer Music Conference*, Perth, 2013.

[7] B. Garbet, “Wait for Me, Daddy,” Calgary, 2015, composition for Wind Ensemble.

[8] J. O’Callaghan, “Bodies-Soundings,” Montreal, 2014, Acousmatic composition.

[9] D. Litke and K. Hamel, “A score-based interface for interactive computer music,” in *Proceedings of the International Computer Music Conference*, 2007.

[10] K. Hamel, “NoteAbility, a comprehensive music notation system,” in *Proceedings of the International Computer Music Conference*, 1998, pp. 506–509.

[11] ——, "NoteAbilityPro II Reference Manual," http://www.Opusonemusic.net/Helpfiles/OSX/Help/Default.html, 2016, [Online; accessed 27-April-2016].

[12] D. Zicarelli, "Max/MSP Software," 1997.

[13] M. Puckette, in *Pure Data*, Hong Kong, 1996.

[14] M. Ritter, "Insomniac's Musings," Vancouver, 2011, composition for Piano and Interactive Electronics.

[15] K. Hamel, "Touch," Vancouver, 2012, composition for Piano and Interactive Electronics.

[16] M. Ritter, "IX. Reach," Vancouver, 2014, composition for Piano, Interactive Electronics, and Gesture Tracking.

[17] J. Schacher and P. Kocher, "Ambisonic Externals for MaxMSP," https://www.zhdk.ch/index.php?id=icst_ambisonicsexternals, 2014, [Online; accessed 27-April-2016].

[18] T. Machover and J. Chung, "Hyperinstruments: Musically intelligent and interactive performance and creativity systems," 1989.

[19] G. Marentakis and S. McAdams, "Perceptual impact of gesture control of spatialization," *ACM Transactions on Applied Perception (TAP)*, vol. 10, no. 4, p. 22, 2013.

Approaches to Real Time Ambisonic Spatialization and Sound Diffusion using Motion Capture

Alyssa Aska
University of Calgary
Calgary, Alberta, Canada
alyssa.aska@ucalgary.ca

Martin Ritter
University of Calgary
Calgary, Alberta, Canada
martin.ritter@ucalgary.ca

ABSTRACT

This paper examines the use of motion capture to control ambisonic spatialization and sound diffusion parameters in real time. The authors use several software programs, developed in Max, to facilitate the gestural control of spatialization. Motion tracking systems using cameras and peripheral devices such as the Leap Motion are explored as viable and expressive means to provide sound localization. This enables the performer to therefore use movement through personal space to control the placement of the sound in a larger performance environment. Three works are discussed, each using a different method: an approach derived from sound diffusion practices, an approach using sonification, and an approach in which the gestures controlling the spatialization are part of the drama of the work. These approaches marry two of the most important research trajectories of the perfor-mance practice of electroacoustic and computer music; the geographical dislocation between the sound source and the actual, perceived sound, and the dislocation of physical causality to the sound.

1. INTRODUCTION

This paper explores live spatialization in three current implementations, suggesting that motion capture is an effective and appropriate means for incorporating gestural sound spatialization into electroacoustic and computer music. For the purposes of this paper, the term gesture will refer to physical actions produced by the performer that have some basic trajectory and emotional intent. Such gestural control of spatialization has been explored quite a bit throughout electroacoustic music history; one could look to early efforts such as Pierre Schaeffer's *potentiometrè d'espace* as precedents for gestural control of sound [1], with more recent historical applications including gestural controllers such as Michel Waiswicz's *Hands*, and advancing to include multi-touch interfaces and further gestural control [2, 3]. Needs for control over spatialization continues to increase as concert halls accommodate more speakers and more complex speaker arrays. There are two primary ways sound can be perceptibly spatialized, with variances existing between each and on a continuum. The first is that of sound diffusion, a technique originating from the *GRM* involving the placement of a sound within a multi-channel system [4]. The second concept is sound choreography, which involves spatial trajectories and placements existing as a primary component of the composition [5]. The term "sound choreography" was firmly established at *IEM* in Graz during a 2010-14 research project entitled "The Choreography of Sound", and is therefore quite new. However, distinction between diffusion and choreography of sound is essentially that one uses the alteration of audio signals to create an illusion of sound movement, whereas the other actually focuses on using a digital mechanism to *localize* the sound in space, and then send it to a speaker. Sound diffusion has been primarily associated with acousmatic music performance practice, whereas sound choreography is more associated with live performance of electronic music (although it could apply to acousmatic music as well). This paper explores the gestural control of spatialization using both sound diffusion and sound choreography, as well as a third approach that lies some-where between the two.

Gestural control of spatialization has been executed many different ways before; in fact, one could argue even that the movement of a fader is a performed gestural action. However, for the purposes of this paper, the term gestural control will apply to those systems, which have been developed specifically with gesture of a performer in mind as the primary driving force between data retrieval and output. Jan Schacher, for example, undertook extensive research on gestural control of 3D sound, which he discussed at length in his 2007 paper "Gesture Control of Sounds in 3D Space". [6] describes a group of modules that can be used control sound in a 3D environment using gesture. However, what differentiates this system from that of the authors is that physical interfaces are used to track gesture in Schacher's software, and that each of the systems described in this paper was designed, at least originally, with a very specific performance practice intention, unlike the generalized systems he discusses. Schacher distinguishes two modes of interaction, *top-down* and *bottom-up*. The former mode involves the performer having direct control over the properties of the sound, whereas the latter enables the performer to interact with a sound that has its own physical properties within the virtual space. The systems discussed within this paper explore both structures described by Schacher, the first system using a *bottom-up* approach, and the final a *top-down*.

Prior research has also been conducted regarding the perception of spatialization relating to gesture. Marentakis

and McAdams studied the effects of gestural spatialization on the audience and found that visual cues improve perception of spatial trajectories, but that such cues result in audience members focusing more on the visual stimuli than the aural [7]. This could be considered a negative for the use of gestural control, especially for music in which the intention is to strip visual stimuli and focus as much as possible on the sound. However, it is a force that could also be harnessed for dramatic or narrative intent, either by increasing the focus on the visual with gestures, or by obscuring the focus on the visual by removing gestures. This paper focuses on using gestures to spatialize sound that have (deliberately) varying degrees of visibility from the audience's perspective. As long as the intent is clear, all are acceptable as meaningful performance techniques.

2. AAAMBI SOFTWARE

All of the approaches that are explored in this paper involve the use of a software module, developed by the author in Max [8, 9], designed to interface with the ambisonics software developed at *ICST* by Jan C. Schacher and Phillipe Kocher [10]. This software, AAAmbi, provides a simple and intuitive graphical user interface intended for use by composers and performers. It allows the user to easily localize sound using both Cartesian and polar coordinates by connecting modules that are designed to send and receive messages from each other. In addition, users can group points, and automate panning features with these easy to use interfaces. AAAmbi has a modular and infinitely expandable design. The user is allowed to specify the number of inputs and outputs to the bpatcher as arguments. This number can be locked and the bpatcher embedded so that connections between the module and other objects are not lost. There is also a user friendly edit panel which can be used to specify speaker configurations, including the azimuth, elevation, and distance of each speaker output. These configurations can be saved and recalled later, which is extremely useful for users who need to rehearse and perform in multiple spaces. This simple dynamic control of input and output arrangement was the original intent of AAAmbi, and the early modules were basic in design and very general in use. However, as new works were created and differing spatialization needs arose, new software modules were created to connect with the AAAmbi software. Three such implementations are discussed below.

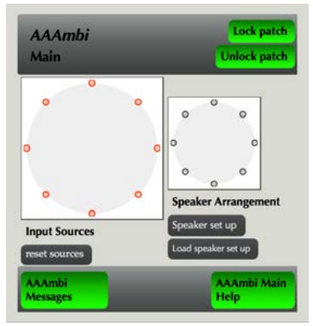


Figure 1: AAAmbi bpatcher.

3. LIVE DIFFUSION USING MOTION TRACKING

The first approach discussed in this paper involves the live diffusion of a stereo work using computer vision tracking in Max. Jean-Marc Pelletier's Computer Vision toolbox objects for Jitter [11] were developed into specialized tracking modules by the author. These tracking modules are easily linked and serve many functions; for the purposes of the diffusion system, the authors use modules that track bounding rectangles of movement within specified areas; the derived data is then used to determine spatial location and speed of movement. This diffusion software was designed for the performance of the specific piece discussed below, but can be used to diffuse any stereo file, if desired.

3.1 Why motion tracking?

With so many easy and accessible systems available, including analog and digital mixers and other devices, it is potentially questionable why one would use motion tracking to diffuse a piece in real time. While there has been some research involving the creation of new diffusion systems that provide a more feasible environment when numerous faders are required, such as the M2 developed by James Mooney, Adrian Moore, and James Moore in Sheffield [12], even such newer systems use a model based on the fader system rather than one on gestural control. However, past gestural controllers required large setup and could potentially take up much space, only adding difficulty to the diffusion process and detracting from the purpose of listening intently. The choice to diffuse this piece using motion arises from research into the performance aesthetics of electroacoustic music, and the particular pieces of hardware used (cameras, Leap motion) were selected based on their accessibility; both are relatively cheap, easily available anywhere, and have regularly updated and improved software and hardware interfaces.

Marko Ciciliani, in his 2014 paper "Towards an Aesthetic of Electronic-Music performance practice" [13], described two models of performance aesthetics, which he termed centripetal and centrifugal. These models are based on the measurement of how visible and central the performing body is to the sound source. Centripetal models consist of those in which the performer is the focal centre and whose movements are very perceptibly linked to the sound. Centrifugal performance, on the other hand, includes works in which the sound enacting body is removed or obscured from vision. Acousmatic works, such as the author's City of Marbles, are by their very nature centrifugal if using Ciciliani's classifications; the sound sources are intended to be unseen by the audience, and traditional diffusion performance would place the composer or other diffusing body hidden from view behind a mixer in a dark room. Using visible gestural diffusion in an acousmatic work marks a big change to the performance practice; the sounds themselves still have no visible source, but a human performer is present, centrally located, and present during performance. For City of Marbles, this is a programmatic intention. One of the primary sources of the work is a voice speaking the Latin phrase: "Marmoream se relinquere, quam latericiam, accepisset," a quote by Augustus Caesar that translates into "I found Rome a city of bricks and left it a city of marbles."

Additional recordings consist of a marble being dropped into various objects, and cicadas captured in Rome at the Palatine hill. The role of the sound diffuser is intended to be central and imposing because the quote itself infers creation and manipulation of several sources by a singular body. However, the agents that produce the actual sound sources are not visible during the performance. Performance of City of Marbles, therefore, uses visible gestural control of spatialization to both foreground the link between gesture and sound, and the obvious lack of visibility of the sound producing agent.

3.2 Tracking of motion to diffuse sound

As mentioned earlier, this system uses computer vision in Max to track the movement of a performer, which translates the data into messages for AAAmbi. The camera outputs a frame, which is then inverted on the horizontal axis (for the purposes of easier visual feedback for the performer) and split into ten separate matrices to be separately analyzed. The first four matrices correspond to faders controlling the output to channels $1, 3, 5$, and 7, and the final four the output to channels $2, 4, 6$, and 8. The matrices in the middle are not used; the division into ten components allows the body to be cut from the processed image, preventing unwanted fader movement. Two implementations from the MR.jit.toolbox [14] are used to analyze the motion: MR.jit.motionComplex and MR.jit.bounds.
MR.jit.motionComplex is used to obtain general information about the amount of motion between successive video frames. It allows the user to specify motion smoothing, binary thresholding, morphological operations, and motion thresholding. The output consists of a binary frame, which can be used for further analysis (see below) and the current total active pixel count. The analyzed frame is then sent to MR.jit.bounds, which calculates a region of motion. These values are used to calculate the area of the motion, horizontal size, vertical size, and velocity and direction of both horizontal and vertical values.

The output is then used to move the sound sources around the performance space. The horizontal parameters are analyzed successively to capture delta values, and these delta values are subsequently used to control the Cartesian coordinates of the sound in space. Once the delta value exceeds a certain threshold, gates are opened within Max, which allow the delta value to control the "Y" parameter on a Cartesian plane, and the X (horizontal) value itself to control the "X" value on a Cartesian plane. The vertical movement data is analyzed in a similar manner, and the resulting delta values control the raising and lowering of virtual faders in the software. These faders are selected using the matrix splitting process described above. Therefore, the performer's hand location in horizontal space selects which fader to affect, and vertical hand/arm movement subsequently modifies this fader value. A high vertical velocity with an upward motion, for example, will result in a strong increase of gain for the fader(s) selected by horizontal positioning. This gesture control system therefore spatializes the sound in two ways: 1) the input sources are moved using horizontal movement data, and 2) the perceived location of sounds is affected by using the vertical movement data to increase or decrease the volume that is output to specific speakers. This second method is much like the original approaches to panning, or the diffusion technique. Since it actually takes quite a bit of motion to trigger the movement of the sounds, the spatialization primarily uses the raising and lowering of virtual faders to diffuse the sound, with some areas that allow for input sources to be moved for an emphatic effect.

3.3 Perception – visual cues in acousmatic music

This piece was initially described as an acousmatic work, and it is such when performed in a dark room on two stereo loudspeakers (or many loudspeakers in stereo pairs). However, when performed with the diffusion software, there is a visual performance element present that changes the performance aesthetics of the work. Gestural control of spatialization has been proven to enhance the audience's awareness of spatial trajectories (as mentioned briefly above), but to also decrease their focus on the sound itself. This runs contrary to the philosophy of acousmatic music, which is derived from the Pythagorean concept of listening to a lecturer behind a wall to focus solely on the sound (this is also where acousmatic music gained its name). City of Marbles, however, uses this live diffusion to illuminate a programmatic element: the narrative of the piece is about an individual creating something (in this case a city), with all of the development controlled by this being. Therefore, the diffuser in this case acts as metaphor for Augustus Caesar, controlling all of these elements to "create" a "city of marble". The lack of visibility of the sound sources is also indicative of the way that great civilizations and cities are created; the "workers" are essentially unseen throughout history, with the ruler of the civilization essentially credited for growth and construction.

4. PIANO HAMMER TRACKING

The second approach to spatialization discussed in this paper involves using a camera to track the mechanical movement of the hammers inside a piano, and software to determine the amount and location of hammer movement. This data is then transmitted to a module called AAAmbiPianoHammers, which translates the hammer movement information to messages that can be sent to AAAmbi, ultimately altering the location of the sound.

4.1 Precedents -IMuSE

The Integrated Multimodal Score-following Environment (IMuSE) was a project at the University of British Columbia, Vancouver, Canada [15, 16]. The system was primarily designed to aid in the rehearsal and performance of score-based interactive computer music compositions. IMuSE incorporates several different software components such as NoteAbilityPro [17, 18] for the notated score (both the traditional score for the performer as well as the score for the electronics) and Max/MSP or pd [19] for the performance of the computer generated sounds as well as the analysis, matching, and networking of the tracking data to NoteAbilityPro. In order to track and score-follow a piece of music, a performance of said piece has to be recorded and the tracking information obtained aligned to the score in NoteAbilityPro.

Tracking data may include any discretely sampled data and the system has been tested with information obtained from accelerometer data, pitch tracking information, amplitude envelopes, spectral information (e.g. spectral brightness), and visual tracking using cameras, the Kinect [20] and the LeapMotion device [21–23]. A variety of instruments have been studied for tracking during and after the project's official timeframe, which include: Viola, Cello, Clarinet, Trombone, Piano, and Accordion.

4.2 Tracking of Hammers

As mentioned above, the authors' approach in tracking the hammer movement is to capture the mechanical action of the instrument, rather than the physical action of the performer itself. This is a direct extension of the research carried out during the piano portion of IMuSE, where auxiliary movements by the performer would on occasion interfere with the tracking as the head or shoulders would enter the tracking area. Capturing the mechanical action inside the piano can eliminate such obstructions. The current version of the software allows the user to crop and rotate the incoming video stream before it is analyzed. This dramatically increases CPU efficiency, as large portions of the video frame do not include any information of importance. This cropped frame is then subdivided into customizable tracking regions. One region per hammer is the default configuration. Each re-gion is then analyzed for movement and if a (user-) defined threshold is exceeded, the region is marked as active, meaning a note was likely played. A list of ON-notes is compiled for each analysis frame and sent to Max for use in the above described software modules.

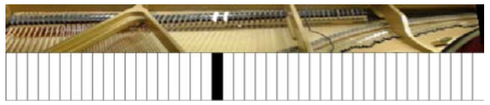


Figure 2: Top: Cropped and rotated image with superimposed movement data (white); Bottom: tracking region divided into 44 discrete areas with currently tracked region in black.

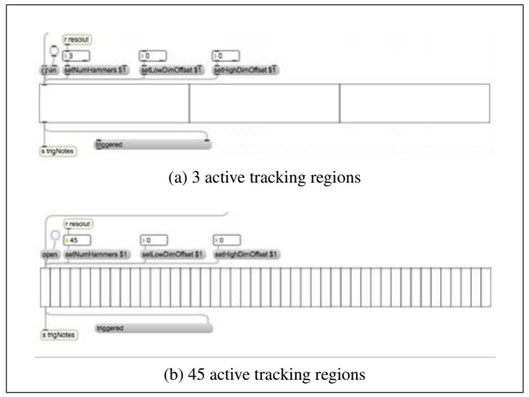


Figure 3: Customizable tracking regions. (a) = 3 active tracking regions; (b) = 45 active tracking regions;

4.3 Application Purpose

Tracking the hammers of a piano and subsequently applying this data to spatialization mappings allows for two primary functions: 1) the frequency space of the piano (i.e. low to high) is translated into spatial trajectories, which enables a perceptible aural correlation between the pitch and spatial location, and 2) tracking inside of the piano removes extra-musical performance gestures, such as seat adjustment, expressive movement, and page-turning, from the motion tracking data. This in turn makes the data more stable and strictly connected to the piano's sounding results. This was the initial intention of the software, which was designed for use in a piano trio by the author that used performance practice as a compositional technique [24]. The piano trio contains motion tracking of all of the instruments, but the tracking data derived from the violin and cello performance is used to modify audio effects. The tracking data from AAAmbiPianoHammers is used to spatialize the piano sounds, and also sounds of prerecorded bells that play back. This is an attempt to connect the bell sound source, which is not visible, with tracking of a mechanism that is not visible (the hammers), but occurs as the result of a visible action (the performer playing the piano). The difference in the relationship between gesture and sound production of a cello and a piano is also pertinent to the choice of hammer tracking. The gesture a cellist makes, for example, occurs right at the physical location that produces sound; a bow is dragged across a string, which then vibrates to produce sound. There is no disconnect between the bow and the hand. A piano is somewhat different in that the performer strikes keys, which are then used as levers to enact hammers to strike strings. There is a visual discon-nect between gesture and sound; the frame of the piano obstructs the hammers visually. What we are seeing relates to the sound, but not exactly. If a cellist makes micro-movements in his hand as a result of nervousness, this will translate into the sound, as the bow will make micro-movements. This is not the case for the piano, and this presents another reason why the authors have chosen to implement the software in this way.

5. USING THE LEAP MOTION TO CREATE A GESTURAL NARRATIVE

Finally, the use of dramatic and/or programmatic gestural movement to control spatialization is discussed. Fayum Fragments, part of a larger poly-work by the author for soprano, flute, cello, piano, and electronics, uses a Leap Motion device to capture the gestures of the vocalist. This gestural interaction is primarily used throughout the work to determine the overall form and narrative, as the structure of the work is aleatoric and dependent on the singer's gestures to advance each section. The Leap Motion also serves the purpose of triggering and spatializing sounds, and it is this particular use of the Leap Motion that will be discussed here.

5.1 Background – MRleap

The *MRleap* object [25] serves as an interface between the Leap Motion USB peripheral device and Max/MSP. It was created specifically to give the user the ability to precisely enable and disable the device's over 90 different gestural data acquisition options. This is extremely useful to performers and composers as it allows them to choose which data streams to apply to audio, visual, or control parameters and avoid unwanted CPU usage by disabling unwanted functionality.

5.2 Using a glyph system for gestural notation

Fayum Fragments uses a system of twelve graphic glyphs the vocalist must interpret with her hands, which loosely translate into gestural movements with certain emotional intent. This is the system the vocalist uses to interface with the Leap Motion throughout the work. The Leap primarily controls the playback of sound files during performance; motion triggers the onset of a sound, and the shape of the motion affects the parameters of the sound as a continuous control. The spatialization of these sounds works in much of the same way, with the starting location of each sound selected and distributed immediately when triggered. The movement shape determines small changes in spatialization, such as the amount of azimuth distance between sources, and the distance of the sources from the centre. Additional processes, such as filters, are added for subtle localization cues.

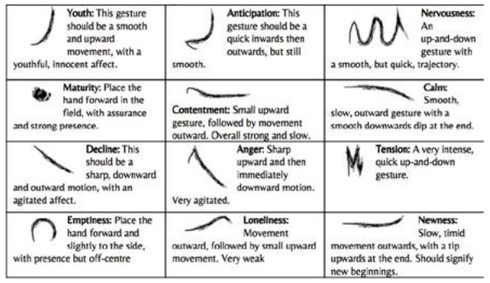


Figure 4: Glyph table showing glyphs and their meanings.

5.3 Gestural data and spatialization

There are only a few parameters of the Leap Motion that are translated into sonic control information during the Fayum Fragments: X, Y, and Z-positions of the hands, and the velocity of movement of the hands. This provides a very simple interface, but with effective musical results. The velocity parameter is used to trigger the sample, and as a result, triggers a randomly selected starting location as a centre point. The continuous X value then controls the degree of circular spread between the sources by modifying the distance value between them. The continuous Z control affects distance from the centre, and Y is used to add a very subtle low pass filter for distance cues. The reason that the trajectory to the starting location is immediate, giving the initial idea of a more stationary localization model, is due to the nature of the samples themselves, which consist of spoken and sung Greek word. These vocal sounds are much more effective and believable when triggered at stationary locations because humans are generally relatively stationary from the listener's perspective when they are speaking. However, once the sounds are triggered, the shape of the gesture allows for very small adjustments in movement. These minor changes in spatialization that coincide with the gestural shape create subtle sonic effects, however, they provide an effective and meaningful link between gesture, spatialization, and sound processing.

6. CONCLUSION

Sound diffusion and choreography using gesture are not novel concepts; systems have been put in place to achieve both, although the gestural control attempts have primarily used physical controllers and sensors using knobs and sliders. Some of the constraints of using such gestural systems purely for diffusion, such as lengthy setup and bulky extra material, are removed by using motion control systems that do not require external hardware controllers. The drawback to using such systems is that motion capture does not necessarily contain the amount of fine control provided by physical sensors that track gesture. 3d tracking systems could be implemented for finer control of parameters, and area consideration for future development. However, for the purposes described in this paper, which involve very general spatialization for mostly artistic and dramatic effect, the tracking of motion using these systems has been effective. Additionally, as the works described in the paper are all concert works intended for multiple performance, accessibility of technology was a large consideration. Thus far, all of the systems have been tested successfully, from technical and aesthetic perspectives, in a controlled lab environment. The diffusion system will be tested in a live performance summer 2016, and the other two systems will be tested in live performance and performance workshops throughout late 2016, culminating in my thesis performance in early 2017.

7. REFERENCES

[1] P. Schaeffer, F. B. Mâche, M. Philippot, F. Bayle, L. Ferrari, I. Malec, and B. Parmegiani, *La musique concrète*. Presses universitaires de France, 1967.

[2] M. Waisvisz, in *The Hands: A set of remote midi-controllers*, 1985.

[3] B. Johnson, M. Norris, and A. Kapur, in *Diffusing Diffusion: A History of the Technological Advances in Spatial Performance*, 2014.

[4] M. Battier, "What the GRM brought to music: from musique concrete to acousmatic music," *Organised Sound*, vol. 12, no. 03, pp. 189–202, 2007.

[5] G. Eckel, M. Rumori, D. Pirro, and R. González-Arroyo, *A framework for the choreography of sound*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2012.

[6] J. C. Schacher, "Gesture control of sounds in 3d space," in *Proceedings of the 7th international conference on New interfaces for musical expression*. ACM, 2007, pp. 358–362.

[7] G. Marentakis and S. McAdams, "Perceptual impact of gesture control of spatialization," *ACM Transactions*

on Applied Perception (TAP), vol. 10, no. 4, p. 22, 2013.

[8] D. Zicarelli, "Max/MSP Software," 1997.

[9] A. Aska, http://www.alyssa-aska.com/software/aaambi, 2016, [Online; accessed 27-April-2016].

[10] J. Schacher and P. Kocher, "Ambisonic Externals for MaxMSP," https://www.zhdk.ch/index.php?id=icst_ambisonicsexternals, 2014, [Online; accessed 27-April-2016].

[11] J.-M. Pelletier, http://www.ajmpelletier.com/cvjit, 2016, [Online; accessed 27-April-2016].

[12] J. Mooney, A. Moore, and D. Moore, "M2 diffusion: The live diffusion of sound in space," in *Proceedings of the International Computer Music Conference 2004*. International Computer Music Association, 2004.

[13] M. Ciciliani, "Towards an Aesthetic of Electronic-Music Performance Practice," in *Proceedings of the International Computer Music Conference*, 2014.

[14] M. Ritter, http://www.martin-ritter.com/software/maxmsp/mr-jit-toolbox, 2016, [Online; accessed 27-April-2016].

[15] M. Ritter, K. Hamel, and B. Pritchard, "Integrated multimodal score-following environment," in *Proceedings of the International Computer Music Conference*, Perth, 2013.

[16] K. Hamel, http://www.opusonemusic.net/muset/imuse.html, 2016, [Online; accessed 27-April-2016].

[17] D. Litke and K. Hamel, "A score-based interface for interactive computer music," in *Proceedings of the International Computer Music Conference*, 2007.

[18] K. Hamel, http://www.opusonemusic.net, 2016, [Online; accessed 27-April-2016].

[19] M. Puckette, in *Pure Data*, Hong Kong, 1996.

[20] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[21] http://www.leapmotion.com, 2016, [Online; accessed 27-April-2016].

[22] M. Ritter and A. Aska, "Leap Motion As Expressive Gestural Interface," in *Proceedings of the International Computer Music Conference*, Athens, 2014.

[23] ——, "Performance as Research Method: Effects of Creative Use on Development of Gestural Control Interfaces," in *Proceedings of the Practice-Based Workshop at NIME'14*, London, 2014.

[24] A. Aska, "Performance practice in electroacoustic music as approach to composition: an examination through two recent works," University of Calgary, Tech. Rep., 2015.

[25] M. Ritter, http://www.martin-ritter.com/software/maxmsp/mrleap, 2016, [Online; accessed 27-April-2016].

Big Tent: A Portable Immersive Intermedia Environment

Benjamin D. Smith
Department of Music and Arts Technology, Indiana University-Purdue University Indianapolis
bds6@iupui.edu

Robin Cox
Department of Music and Arts Technology, Indiana University-Purdue University Indianapolis
robcox@iupui.edu

ABSTRACT

Big Tent, a large scale portable environment for 360 degree immersive video and audio artistic presentation and research, is described and initial experiences are reported. Unlike other fully-surround environments of considerable size, Big Tent may be easily transported and setup in any space with adequate foot print, allowing immersive, interactive content to be brought to non-typical audiences and environments. Construction and implementation of Big Tent focused on maximizing portability by minimizing setup and tear down time, crew requirements, maintenance costs, and transport costs. A variety of different performance and installation events are discussed, exploring the possibilities Big Tent presents to contemporary multi-media artistic creation.

1. INTRODUCTION

Large-scale immersive environments serve as compelling venues for contemporary artistic exploration and research. These activated spaces allow creators to treat the environment as an instrument, using the walls as an interactive visual canvas coupled with surround audio systems (see [5]). However the spaces are typically expensive to create, have limited accessibility, and come with elitist stigmas. These aspects restrict audiences and constrain many musicians and artists in their attempts to explore aesthetic possibilities of fully immersive spaces.

Big Tent presents a new approach, seeking to provide a portable, accessible environment for creators and audiences alike to experience inter-media art and music (see Fig. 1). Through scale and portability, the design brings possibilities of 360° surround video and audio to nearly any location, for group experiences and performances, while serving as a reliable environment for artistic exploration and research. This instrument is aimed at enabling a diversity of events, aesthetic orientations, and genres.

2. BACKGROUND

One of the primary requirements of research in any field is replicability, serving as a basis for validating and sharing findings. Aesthetic explorations of Big Tent are intended to be highly replicable, hoping to make every examination repeatable and shown again and again, just as a physical painting or a composition for orchestra. Much work in cutting-edge experimental areas either intentionally or incidentally denies or fails to honor this requirement. In fact, aesthetic research through artistic expression, which is the domain of all creative artists and musicians, can greatly benefit from embracing this model. Supporting experimental replication strengthens the field as a whole and enables more directed and connected creativity and research.

Figure 1. Big Tent performance event.

The Big Tent enables replicability by providing both a stable and predictable apparatus and technical parameters to ground artist explorations in the chaotic domain of digitally powered mixed media. Physically, Big Tent is a 40-foot diameter ring of 8 projection screens, standing 12 feet tall, with a projectable surface 128-feet around (figure 2). This is augmented with 8 channels of surround audio, and 8 channels of HD video to fill the surface. The entirety is driven by audio/visual software providing a flexible interface so artists may work and play with the environment in a creatively supportive fashion.

Artistic expression exploiting digital and computing technology has become ubiquitous with the relatively recent advent of personal computing and mobile computing. Innovative creators around the world continuously push limits, on what is potentially one of the biggest frontiers of expression today. Our laptops, tablets, and smart phones are immensely expressive tools with an artistic reach limited only by the artists' conceptual abilities.

However, the myriad approaches to tackling the aesthetic possibilities of digital technology make the codification of the field extremely difficult. Inter-media art is not only new, lacking the heritage of more traditional art practices, it is also extremely diverse and encompassing, involving elements from many other art forms. Even comparing two similar works or artists is difficult due to

the differences in setup and technology employed in every case. It is as if every artist, more fundamentally, is inventing their own tools, equivalent to a painter making paintbrushes from scratch, fabricating their own canvases, and looking for the newest pigments with nearly every work. Learning from this wealth of experimentation and finding best practices is difficult.

In the field of music we look to the origination of the violin family of instruments and the aesthetic grounding this afforded as a model solution. Prior to the modern violin, string instruments were extremely diverse, with varied capabilities, tunings, playing techniques and expressive ranges. While composers worked with these instruments of diverse musical abilities the dissemination of pieces was difficult. Once the violin as we know it began to spread, with its standard tuning, playing techniques, and pitch range, composers had effectively found a uniform canvas on which to work. When Mozart wrote a piece, he could be confident it would sound the same played in disparate locations such as London and Vienna. The consistency in the instrument, the musical toolset, allowed composers to share, explore, and learn from one another's experiments (i.e. pieces), effectively mapping out the capabilities of the violin and its expressive potential (a process that continues today).

Big Tent aims to take steps in this direction as a modern music-technology instrument, providing a consistent canvas for inter-media artists to explore and work on. Due to its portability, being usable in any space with a sufficient footprint, indoor or out, and ease of construction, requiring two hours for a team of four to set it up, Big Tent may be erected as a presentational venue in both traditional and unconventional circumstances (from concert halls and art museums to parks and parking lots). In the same way that a violinist or dancer may perform in any setting they desire, Big Tent allows artists to play with location and take the instrument to the preferred environment or audience.

Other environments have been created with similar technology, but none with the portable arts-research laboratory aims of Big Tent. Scientific virtual reality (VR) systems are one such example, perhaps best exemplified by NASA's HIVE environment [1, 3], a portable VR display system. Yet, the HIVE focuses on solving different problems, being a single user experience, necessitating a fixed viewer orientation, and being prohibitively expensive to construct. The Allosphere at UCSB [4], a large-scale facility for advanced research in immersive environments provides a complete sphere of video and audio several stories tall, existing in a dedicated building. However, this space is not at all portable or flexible in application.

Artists who have created their own environments for their work include Bill Viola (who frequently works with multiple video and audio sources in fixed gallery settings), and Maurice Benayoun and his Cosmopolis (2005) [2]. Similar in concept to Big Tent, Cosmopolis involved a ring of 12 projection surfaces with surround audio, yet the design was unique to this single work with specifically tailored interaction points and not easily transferred to new locations or other pieces.

3. DESIGN PRIORITIES

The design goals for Big Tent were to create an aesthetically neutral venue for audiences of up to 60 people, supporting a broad stylistic range of music, dance, and intermedia art expression. It also must accommodate different modes of performance and communication in many different contexts, such as concerts, installations, and interactive works, presented within conventional facilities (e.g. museums and concert halls) and non-conventional spaces (e.g. parks, gyms, and shopping centers).

With the primary goal of portability, three issues are at the forefront of consideration:

1) Ease of transport: minimize equipment weight, volume, and packed footprint;
2) Ease of setup: minimize number of crew and time required to build the Tent on location;
3) Ease of tear down: minimize time to deconstruct the Tent and load it into a vehicle for transport;
4) Ease of maintenance: minimize operating costs and replacing broken equipment.

The target cost points were no more than 4 technical crew working on setup and teardown, requiring no more than 3 hours before and after an event, and using commercially available components (for cost and easy replacement).

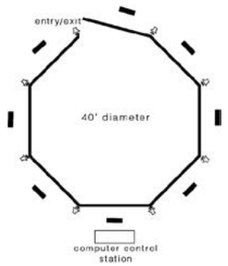


Figure 2. Big Tent layout and floor plan.

4. IMPLEMENTATION

4.1 Frame

The physical structure of the Tent is designed to balance robustness against ease of setup and transport, all while minimizing cost (Fig. 2). The 128-foot octagon framework supporting the screens is a hybrid of ¾" steel pipe and tripod lighting stands. The light stands are off-the-shelf products capable of a 12' height and load bearing of 77.1 lbs. The screens hang from a top truss ring, constructed of steel pipe resting on top of each stand. Each junction point is built out of pipe fittings allowing for any arbitrary angle, enabling both flexibility in setup, and possible configurations of the Tent in asymmetrical octagonal shapes (to account for environmental obstacles, non-square spaces, or artistic preference). Screen tensioning is accomplished entirely by elastic ties at the junctions. It was determined that this alone provided sufficient tensioning to eliminate most wrinkles, alleviating the need to add piping at the bottom of screens and subjecting the setup process to additional screen stretching, which in turn further reduces transport costs and setup time.

In order to keep the interior of the Tent completely free of wires or other visible pieces of equipment, rear video projection is used for all of the screen surfaces. The projectors selected for Big Tent are ultra-short throw Epson projectors. Each is capable of 3100 lumens and has a throw ratio of 0.27:1 (e.g. for every 0.27 feet of distance it can cover 1 foot of screen). Optimally, these projectors fill a 16' wide screen from just a distance of 4.32 feet. This affords Big Tent to be completely setup within a 50x50-foot space, yet retaining a full 40-foot diameter internal area with less than a 5-foot ring behind the screens for equipment.

The projectors are placed on the floor and adjusted manually to fill the screens. In order to remove hotspot glare (i.e. where the projector bulb is seen through the screen) projectors cannot be directly behind a screen relative to the viewer's eye. Thus the screens are raised 36-inches off the floor (bringing the top rim of the Tent to 12-feet). This allows projectors to aim upward and prevent visible glare to viewers inside the tent. A black cloth skirt was added below the screens to eliminate visibility of the projectors under the screens and provide a further element of definition to the temporary Tent walls.

4.2 Video

All of the projected video content is distributed from a single Mac Pro (with 12, 3.5ghz CPUs) to the 8 projectors, each displaying 1280x720 pixels. The operating system treats the 8 screens as an extended desktop, creating a single 7 megapixel surface. This allows any Mac multi-media software to use the entire projectable area.

Despite the theoretical throw ratio of the projectors, the selected model only provides plus-minus 5 degrees of vertical key-stoning, which alone is insufficient to account for the necessary placement of the projectors (below the screens). Therefore additional key-stoning is performed digitally in software (using Qlab), to provide uniform pixel size.

4.3 Audio

A robust conventional 8 channel audio system of 280 watt speakers installed in the Tent provides fully surround audio. A sub-woofer is also placed outside the Tent near one of the screen junctions and the eight speakers are set at the base of each junction with a slight upward angle. While this is not acoustically ideal, it greatly expedites setup and teardown time and assists to minimize the visual presence of the speakers inside the Tent (see Fig. 3, 4).

Audio/video synchrony is maintained simply by running both subsystems on a single computer, enabling any multichannel capable application to use the Tent. The aforementioned Mac Pro drives the audio and delivers the 7 mega pixels of video (guaranteeing a minimum frame rate of 15 frames per second).

5. PERFORMANCES

Figure 3. Interactive video and live musical performance inside Big Tent.

Testing the functionality and capabilities of this new hyper-instrument, and exploring the aesthetic potentials, has commenced through several public live performance events and installations. These events and individual artistic pieces worked with a variety of media and interaction models in an attempt to discover the potentials of Big Tent as well as identify technical issues and limitations. The variety of sources and approaches comprise the following modalities (used singly and in combination):

- One HD video shown simultaneously on 8 screens.
- One HD video shown simultaneously on all screens with different time delays and playback rates.
- One interactively generated animation displayed across all screens (see Fig. 1 & 3).
- One 720p HD video stretched across all 8 screens.
- One 2560x720 HD video repeated 4 times over all the screens.
- Many SD videos displayed concurrently in a haphazard fashion on any/all screens.
- Prerecorded surround audio.
- Live acoustic instrument performance.
- Interactive audience driven music.
- Solo dance (see Fig. 4).
- Contact improvisation, audience participation dance.

Original works created and/or adapted for the Tent using these approaches were staged in four multi-hour public concert events, an evening length interactive installation, two hour-long contact improvisation events, and a week long fixed-media installation.

Ambient light was quickly identified as an important factor in Big Tent's performance. Rear projection is very unforgiving of ambient light present at the event location. Near total darkness is required for adequate screen illumination of projected images. In outdoor environments

Big Tent is only viable at dusk and into the night, and when indoors any light source must be low in output and focused away from the screens.

Figure 4. Dance and video in Big Tent.

Big Tent's 40-foot internal diameter was capable of accommodating an audience size of 40 to 60, even with two live performers in the middle of the space. Furthermore, the scale of Big Tent allows one to experience a presentation without the sense of confinement or lack of a peripheral depth of field characteristic of other immersive multimedia environments.

The lack of primary orientation became quickly apparent as audiences were encouraged to walk around the space. Many immersive spaces retain a notion of front and back, akin to conventional concert hall orientation, but the circular nature of Big Tent appears to dissolve this approach. Artists creating works for the space mostly abandoned the notion of primary orientation, repeating sounds and imagery around the whole space.

Audio coverage in the space was very satisfactory, given the 8.1 surround system. However, the placement of speakers at the base of the screen junctions fails to allow discreetly localized sound spatialization. A system of small satellite speakers mounted at head height or slightly above could address this issue.

While only subtle in noticeable effect, given the breadth of these projection surfaces, the general frame rate limitation of about 15fps allows an occasional degrading of continuous motion video projection. However, this design provides a very stable and elegant means of streaming to 8 digital projectors, thus addressing common video production problems of computer equipment costs and synchronization across machines. Working with the screens has been relatively transparent for artists involved thus far.

The transportability and relatively easy setup and breakdown has proven very successful. The shear portability of Big Tent has already shown itself to meet the design expectations of taking multi-media immersive presentations out into the community and away from traditional event settings. Four crew members can unpack and setup the Tent in under 3 hours, and pack it up and load it out in under 1 hour.

6. CONCLUSIONS

One of the primary limitations constraining the use of Big Tent is power consumption. With all components turned on Big Tent requires just under 30 amps of power, which is more than the average single circuit in most US buildings (15 amps is the most commonly available). Thus Big Tent requires two separate circuits or special accommodations to provide the required power for a setup and event.

One of the primary aims with Big Tent is to support out-of-doors events, yet weatherproofing is required to enable this.

Environmental light has similarly been identified as a primary challenge limiting the use of Big Tent. Due to the back projection system employed ambient light from the installation environment bleeds through and washes out the video. Typically Big Tent can only be used after the sun has set or in spaces where all lights can be turned off, relying solely upon the Tent's video projectors for all event lighting. Solutions for this may involve a second exterior tent, made of a heavy, non-light-permeable canvas, which would contain Big Tent and reduce ambient light. Similarly, much higher lumen projectors could combat ambient light, but coming with greatly increased equipment costs.

While the single-computer system configuration comes with certain advantages, it also has limitations. Currently, the most problematic is the video frame rate, which is less than desired. A target of 60 frames per second is ideal, which could be accomplished by synchronizing several computers driving two or three projectors each.

7. ACKNOWLEDGEMENTS

This project is funded by the IUPUI Arts and Humanities Institute and the Deans' Office of the IUPUI School of Engineering and Technology.

8. REFERENCES

[1] Boniface, Andrew. 2015. "T-38 Primary Flight Display Prototyping and HIVE Support Abstract & Summary." In Houston, TX: NASA.

[2] Cubitt, Sean, and Paul Thomas. 2013. *Relive: Media Art Histories*. MIT Press.

[3] "DEVELOP's HIVE: Redesigning and Redefining the 3-D Virtual Environment." 2012. *Earthzine*. August 13. http://earthzine.org/2012/08/13/develops-hive-redesigning-and-redefining-the-3-d-virtual-environment/.

[4] Höllerer, Tobias, JoAnn Kuchera-Morin, and Xavier Amatriain. 2007. "The Allosphere: a Large-scale Immersive Surround-view Instrument." In *Proceedings of the 2007 Workshop on Emerging Displays Technologies: Images and Beyond: The Future of Displays and Interacton*, 3. San Diego, California: ACM.

[5] Yi, Cheng, and Xiang Ning. 2009. "Arts and Technology Alter Each Other: Experimental Media and Performing Arts Center by Grimshaw at RPI." *Architectural Journal* 11: 018.

Gesture-based Collaborative Virtual Reality Performance in *Carillon*

Rob Hamilton
Rensselaer Polytechnic Institute
hamilr4@rpi.edu

Chris Platz
Savannah College of Art and Design
leonplatz@gmail.com

ABSTRACT

Within immersive computer-based rendered environments, the control of virtual musical instruments and sound-making entities demands new compositional frameworks, interaction models and mapping schemata for composer and performer alike. One set of strategies focuses compositional attention on crossmodal and multimodal interaction schema, coupling physical real-world gesture to the action and motion of virtual entities, themselves driving the creation and control of procedurally-generated musical sound. This paper explores the interaction design and compositional processes engaged in the creation of Carillon, a musical composition and interactive performance environment focused around a multiplayer collaboratively-controlled virtual instrument and presented using head-mounted displays (HMD) and gesture-based hand tracking.

1. INTRODUCTION

For as long as computers have been purposed as real-time generators and controllers of musical sound, composers and performers have researched methods and mappings through which performative gesture can intuitively drive computer-based instruments and procedures [1]. Traditional instrumental performance practices, developed over centuries of musical evolution, have by their very nature been based in the physical control of physical interactive systems. While digital music systems have freed musical generation and control from the constraints of physical interaction, there exists a strong desire amongst contemporary composers, performers and researchers to develop idiomatic performance mappings linking musicians' physical gestures to computer-generated music systems [2].

As commercial high-resolution virtual reality systems become commonplace components of an already digitally immersed 21st Century culture, a natural reaction for composers seeking to use rendered space for musical exploration is to look to existing instrumental performance paradigms and gestural mappings to guide interaction models for musical control in VR space. In that light, digital artists and researchers have been exploring modes of crossmodal interaction that allow users to control and manipulate objects in a rendered reality using interfaces and physical interaction models based in their own physical realities.

Figure 1. Live performance of *Carillon* featuring the Stanford Laptop Orchestra at Stanford University, May 30, 2015.

2. OVERVIEW

Carillon is a mixed-reality musical performance work composed and designed to marry gesture in physical space, avatar and structure motion and action in virtual space, and the procedural musical sonification and spatialization of their resultant data streams within a multi-channel sound space. Premiered on May 30, 2015 at Stanford University's Bing Concert Hall by the Stanford Laptop Orchestra [3], *Carillon* allows performers in VR space to interact with components of a giant virtual carillon across the network, controlling the motion and rotation of in-engine actors that themselves generate sound and music.

Visually, *Carillon* incorporates rendered three-dimensional imagery both projected on a large display for audiences to view as well as presented stereoscopically in a Head Mounted Display (HMD). Performers wearing Oculus Rift head-mounted displays view the central carillon instrument from a virtual location atop a central platform, overlooking the main set of rotating rings. Each performer's viewpoint aligns with one of three avatars standing in the virtual scene. Using Leap Motion devices attached to each Oculus Rift headset, each performer's hand motion, rotation and position are mapped to the motion, rotation and position of the hands of their respective avatar, creating a strong sense of presence in the scene. Floating in front of each performer is a small representation of the main set of rings that can be "activated" by touching one or more rings. A hand-swipe gesture is used to expand or collapse the set of rings, and each ring is visually highlighted with a distinct red color change when activated

Sound is generated in *Carillon* procedurally, by mapping data from the environment to parameters of various sound models created within Pure Data. The parameters of motion of each ring - speed of rotation in three-dimensional coordinate space - are mapped to parameters of a model

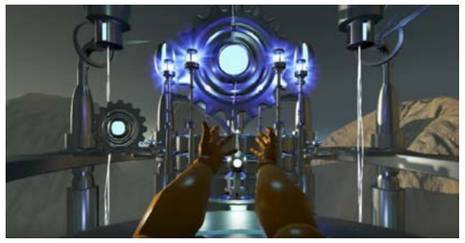


Figure 2. Avatar hands, the large central set of rings and the smaller "HUD" rings to the avatar's left.

of a "Risset bell" (formed using additive synthesis) [4] in Pure Data [5]. Open Sound Control [6] is used to export data from the Unreal Engine 4 [1] and pass that data to . Rules governing the speed of rotation for each ring mediate user gesture and intentionally remove each performer from having precise control over the performance's output. Similarly, by allowing multiple performers within the environment to interact with the same set of rings, the instrument itself and any performance utilizing it become inherently collaborative experiences. During performance, clients or "soloists" connect to a game server and control individual instances of their character avatar within a shared and networked virtual space on the server.

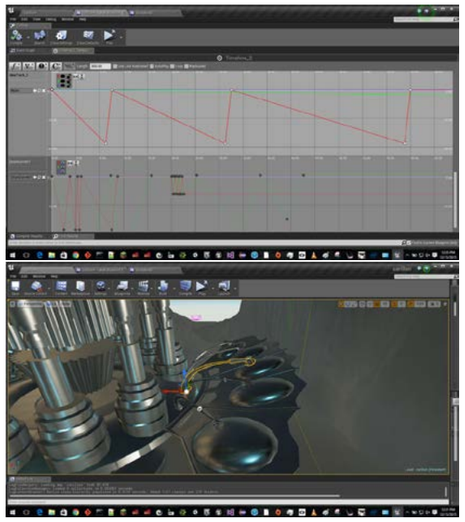


Figure 3. Event timeline (top)in Unreal Engine 4 triggers the motion of OSC-generating bell strikers (bottom).

In addition to the interactive sound-generating processes controlled by performers, a second musical component to *Carillon* is performed by a series of bell-plates and animated strikers attached to the main bell-tower structure. A series of scripts generated in the Unreal Timeline editor drive the animation sequence of each striker according to a pre-composed sequence created by the composer. When triggered by the Timeline, strikers swing and contact bell-plates, triggering an OSC message which drives the bell model in Pure Data. In this manner, composed rhythmic and harmonic material interacts with the live performance gestures controlled by each performer, adding structure to the work while still allowing a great deal of improvisation and interplay during any performance.

An accompaniment to both live soloists and pre-composed bell sequences is provided by a third sound component to *Carillon*, namely an ensemble laptop orchestra layer. Performers control an interactive granulation environment written in ChucK [7] while following a written score and a live conductor. Material for the granulator instrument is generated from pre-recorded samples of percussive steel plate strikes and scrapes, and the striking and scraping of strings within a piano.

[1] Epic Games, https://www.unrealengine.com

3. MODES OF PERFORMANCE

In addition to the original ensemble performance configuration featuring soloists wearing head-mounted displays *Carillon* has been successfully presented in two additional configurations including a solo multi-channel performance and an interactive gallery installation.

Figure 4. Solo performance of *Carillon* featuring two Leap Motion controllers and sixteen-channel audio.

3.1 Solo-performance

One drawback of *Carillon* performances with HMD-wearing performers has been the disparity between the immersive three-dimensional views presented to each performer and the two-dimensional rendering of the instrument/environment presented to the audience. During a number of solo performances of *Carillon*, the performer's in-engine view was presented to the audience, allowing them to see the complete interplay between Leap-controlled avatar arms and the performative gestures driving the musical system (see Figure 4). In this configuration, a Windows shell script triggered both dynamic camera views of the environment as well as the bell sequence. A multi-channel version of the ChucK patch performed by the laptop orchestra was driven by a second Leap Motion controller, with gestures based on hand motion, location and relative placement all mapped to parameters of the ChucK instrument. Output from Pure Data and ChucK instruments were spatialized around a sixteen-channel speaker environment.

3.2 Gallery Installation

As part of the "Resonant Structures" gallery show at Stony Brook University's Paul Zuccaire Gallery, an installation version of *Carillon* was designed and implemented (see Figure 5). Using two Leap Motion controllers and a single display, visitors could control the virtual carillon, listening to their collaborative sonic output over two sets of headphones. Like the solo-performance version, aspects of the installation including the bell sequence and camera views were triggered and controlled by a looping shell script.

Figure 5. Gallery Installation: *Carillon* on display at Stony Brook University's Paul Zuccaire Gallery, 2016.

4. GESTURE AND MOTION

The role of gesture in musical performance can be both structural and artistic. In traditional musical performance, physical gesture has been necessary to generate sufficient energy to put instrumental resonating systems into motion. Musical performers and dancers at the same time convey intention, performative nuance and expressivity through their gestures, shaping the performance experience for performers and audiences alike by communicating without language [8, 9]. In each case, the injection of energy into a system, be it via articulated gesture into a physical resonating system or through a sequence of motion into a choreographed or improvised pattern of action.

In *Carillon* a conscious effort was made to impart physicality into the control of the instrument, itself only existing in rendered virtual space. Hand motions in three-dimensions - respectively pushing into the screen, swiping left or right or moving up or down - inject energy into the selected ring or rings, causing them to rotate around the chosen axis. In this manner, human physical gesture as articulated through the hands and arms is used to mimic the physical grabbing and rotation of the gears of the main instrument. The angular velocity of performer gesture translates into directional rotation speed for the instrument, affecting different sonic parameters for each direction of rotation.

5. PARAMETER MAPPING

To create a tight coupling between the physically-guided rotation of *Carillon*'s central rings and the sonic output that they controlled, parameters representing the speed and current directional velocities for each ring were used to control a Pure Data patch based around Jean-Claude Risset's classic additive synthesis bell model.

For each ring, a set of starting partials was calculated and tuned by hand, to create a set of complementary pitch spaces. The root frequency of each bell model was driven by the speed of rotation in the X or horizontal plane (as viewed by the performer). The amplitude of each bell model was driven by speed of rotation in the Z or vertical plane. A quick gesture pushing into the Y plane to stops a selected ring or set of rings.

To add to the harmonic complexity of the work, the starting partials for each ring were varied for each individual performer, with each performer's output being spatialized to a different set of outputs. In this manner, performers had to cooperate with one another to create or preserve interesting timbral and harmonic structures. As it was not always immediately apparent as to which parameter from each performer was creating a particular desired timbre or harmonic structure, performers had to employ a great amount of listening and exploration to move the performance in desired directions.

6. SYSTEM ARCHITECTURE

The visual and interactive attributes of *Carillon* were developed using the Unreal Engine 4 by Epic Studios, a commercial game-development engine used for many commercial game titles. The Unreal Engine 4 is free to use for non-commercial projects and can be used in commercial projects based on a profit-sharing licensing model. Art assets including 3D objects, animations, textures and shaders were created using industry-standard tools including 3ds Max and Maya by Autodesk.

Within the Unreal Engine, the Blueprint scripting language - a workflow programming language for controlling processes within Unreal - was used to script the interactions between player and environments, the networking layer, and custom camera and player behaviors. External plugins, developed by members of the Unreal and Leap Motion developer communities were used to bind hand-tracking data from the Leap Motion devices to avatar limb skeletal meshes [2] as well as to output Open Sound Control messages from Unreal to Pure Data [3].

6.1 Bell Sequences

A precomposed array of struck Risset bell models is used as an additional compositional element. A sequence of pitches was composed and stored within Pure Data. Notes from this sequence were triggered by OSC messages generated by collisions between rendered hammers and a series of rendered bell-like plates. Collision boxes attached to each striker were scripted to generate unique OSC messages when they collided with each plate, turning each visual artifact into a functioning musical instrument.

[2] https://github.com/getnamo/leap-ue4
[3] https://github.com/monsieurgustav/UE4-OSC

Rather than control the motion of each bell striker from Pure Data over OSC, a novel technique native to the Unreal Engine was explored for this musical control system. Using the Blueprint "Timeline" object, multiple parameter envelope tracks representing the speed of motion and angle of articulation for each individual bell striker can be precisely set over predefined time periods (see Figure 3). Timeline object parameter tracks are typically used to automated variables for specified game entities over a given timeframe. The automation of an OSC enabled interaction, occurring when the rotation of a striker entity is driven by the timeline track to collide with a bell-plate entity, serves as a notated multi-part score wholly contained within the Unreal Engine's internal interface.

6.2 Laptop Orchestra Arrangement

For the premiere performance of *Carillon*, an additional laptop-based performance interface was created for the Stanford Laptop Orchestra. Following a notated score and a human conductor, the ensemble of nine performers controlled individual instances of a software granulator written in the ChucK programming language. Sound for each SLOrk member was produced from a six-channel hemispherical speaker alongside each performer, distributed around the concert stage in a semi-circle. Sound material for the accompaniment was composed using fragments of metallic percussion and string recordings and granulated in real-time by each performer. Gestures notated in the performance score matching temporal cues from the Timeline-driven carillon-bell tracks were performed en masse by the ensemble. In this manner live gesture performed by *Carillon's* soloists is married to both the pre-composed bell tracks as well as the real-time granulated performance by the laptop orchestra.

7. DISCUSSION

As an exploration of rendered space and that space's suitability to sustain performance interactions capable of driving music and sound, *Carillon* has been an extremely successful work. The integration of head-mounted display, hand-tracking sensors and procedurally-generated sound creates a novel yet physically intuitive interaction model that can be learned quickly yet explored to create nuanced sonic gesture. The client-server architecture of the work allows for multiple potential configurations ranging from ensemble to solo performance, from gallery installation to networked "game" play, as shared on the Leap Motion developers website.

Carillon was designed with the specific intent of exploring the nature of collaborative instruments controlled by human gesture while residing in a shared network space. Simple yet intuitive physical gestures as tracked by Leap Motion sensors allow performers to affect changes upon the instruments procedurally-realized timbre, frequency and amplitude at varying scale (from small to large) and in real-time. By presenting *Carillon* to soloist performers using HMDs, the feeling of depth and presence associated with functioning VR devices allows the performer to utilize depth in gesture more accurately and intuitively. The performative yet necessarily collaborative aspects of *Carillon*'s central ring-as-instrument metaphor not only allows each performer to improvise freely, but also adds a level of constraint in each performer's ability to either inhibit or augment one another's gestures. A more complex and articulated musical work is realized through the addition of pre-composed bell sequences and a live-yet-composed laptop orchestra accompaniment.

8. ACKNOWLEDGEMENTS

The design, development and production of *Carillon* has been made possible through generous hardware grants by NVIDIA and Leap Motion.

9. REFERENCES

[1] M. Mathews and R. Moore, "GROOVE: A Program to Compose, Store, and Edit Function of Time," *Communication of the ACM*, vol. 13, no. 12, 1970.

[2] M. Wanderley, "Performer-Instrument Interaction: Applications to Gestural Control of Music," Ph.D. dissertation, University Pierre Marie Curie - Paris VI, Paris, France, 2001.

[3] G. Wang, N. Bryan, J. Oh, and R. Hamilton, "Stanford Laptop Orchestra (SLORK)," in *Proceedings of the International Computer Music Association Conference*, Montreal, Canada, 2009.

[4] C. Dodge and T. Jerse, *Computer Music: Synthesis, Composition and Performance*. Schirmer, 1997.

[5] M. Puckette, "Pure Data," in *Proceedings, International Computer Music Conference*. San Francisco: International Computer Music Association, 1996, pp. 269–272.

[6] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, pp. 193–200, 2005.

[7] G. Wang, "The ChucK Audio Programming Language: A Strongly-timed and On-the-fly Environmentality," Ph.D. dissertation, Princeton University, Princeton, New Jersey, 2008.

[8] S. Dahl, F. Bevilacqua, R. Bresin, M. Clayton, L. Leante, I. Poggi, and N. Rasamimanana, *Gestures In Performance*. New York: Routledge, 2010, pp. 36–68.

[9] A. Jensenius, M. Wanderley, R. Gody, and M. Leman, *Musical Gestures: Concepts and Methods of Research*. New York: Routledge, 2010, p. 13.

Emphasizing Form in Virtual Reality-Based Music Performance

Zachary Berkowitz
Louisiana State University
zberko1@lsu.edu

Edgar Berdahl
Louisiana State University
edgarberdahl@lsu.edu

Stephen David Beck
Louisiana State University
sdbeck@lsu.edu

ABSTRACT

The role of form in virtual reality-based musical performance is discussed using historical precedents and current research examples. Two basic approaches incorporating 3D environments into musical performance are considered: a "static" approach in which the space does not change but is instead explored and interpreted by the performer, and a "dynamic" approach in which movement of the space or objects within the space directly influences or controls the performance of the music.

These two approaches are contextualized through works such as "Poème Électronique" and other historical works of spatial music, with particular attention to the spatial notation methods and mobile forms employed by composer Earle Brown in works such as "December 1952" and "Calder Piece." Through discussion and demonstration of his own compositions "Zebra" and "Calder Song," the lead author explores how Brown's ideas can be developed, re-examined, and re-imagined in virtual space.

1. INTRODUCTION

The relationship between musical form and physical space has long been a concern for composers.

When considering a musical performance in virtual visual space, it is important to consider environment not only as a means for creating a more novel or immersive experience, but also as an essential component of the musical form. Among the ways for incorporating a virtual visual space into musical performance, two basic conceptual approaches can be considered:

1. A "static" virtual space, in which the space does not change but is instead explored and interpreted by the performer, and

2. a "dynamic" virtual space, in which movement of the space or objects within the space directly influences or controls the performance of the music.

Through the history of musical performance in virtual reality, the mobile forms of Earle Brown, and the current work of the authors, the how and why of these two methods for musical performance in virtual space can be better analyzed and understood, and the principles of virtual space-as-form can be brought into practice.

2. PRECEDENTS: MUSICAL FORM AND PHYSICAL SPACE

Consideration for the effect of physical space on musical form can be observed throughout musical history. For example, *antiphony*, the practice of distributing a composition between multiple choirs or other performance ensembles, is a staple of both Western music and music of cultures around the world. In Western music, antiphony can be found (among many other examples) in the polychoral music of Venetian school composers such as Giovanni Gabrieli, the works of Classical composers such as Mozart (e.g. *Notturno* in D major for four orchestras), and the works of European modernists such as Karlheinz Stockhausen (e.g. *Gruppen*) and Bruno Maderna (*Quadrivium*) [1].

Using loudspeakers, 20th and 21st century composers have been able to more easily achieve a similar effect. The multimedia installation *Poème Électronique* (1958) by Edgard Varèse, Le Corbusier, and Iannis Xenakis is a particularly notable example of the intersection of physical space and musical form and explores the idea of a navigable composition. Thus, it can be viewed as a direct precursor to modern methods of composition in virtual space. This connection was made by Vincenzo Lombardo and colleagues, resulting in their reconstruction of the architectural component in virtual reality [2]. The authors of this project state, "The integration of music and image inside a space has been said to make *Poème Électronique* . . . the first modern multimedia event—and, one could argue, an *ante litteram* virtual reality installation" [3].

3. FROM PHYSICAL TO VIRTUAL

The year 1992 was a watershed moment in the development of virtual world musical performance. In 1992, Virtual Reality (VR) pioneer Jaron Lanier provided an early example of musical performance in VR when he staged a live performance with VR-based instruments entitled *The Sound of One Hand*. The program notes for this performance read:

> A live improvisation on musical instruments that exist only in virtual reality [sic]. The piece is performed by a single hand in a DataGlove. The audience sees a projection of the performer's point of view. The instruments are somewhat autonomous, and occasionally fight back. The music changes dramatically from one performance to the next [4].

Also showcased at SIGGRAPH in 1992 was the CAVE system, developed by researchers at the University of Illi-

nois Chicago Electronic Visualization Lab [5]. In a valuable statement on the role of the audience in VR, the authors state in their SIGGRAPH '92 article "One of the most important aspects of visualization is communication. For virtual reality to become an effective and complete visualization tool, it must permit more than one user in the same environment" [5].

One can further consider the recent rise in popularity of the video game streaming service Twitch (`http://www.twitch.tv/`) as part of a movement toward performance-oriented virtual worlds.

Finally, the idea of 3D video game as performance has been explored further in a musical context by artists such as Robert Hamilton, whose works such as *ECHO::Canyon* and several others use game physics and virtual environments to control sonic processes [6, 7]. These exciting new works are helping to deepen understanding of the role of virtual space in music composition and the influence of video game culture on the computer music landscape.

4. COMPOSITIONAL APPROACHES FOR VIRTUAL SPACE

4.1 Static Approaches

Poème Électronique can be considered a static approach to virtual world creation. Sounds and images move around the space, but neither the space nor the objects within the space are moving or changing. The space affects the compositional methodology and the perception of the sounds and images, but the space itself does not serve to create or compose the content. Instead, the space adds a navigable component to pre-composed content.

Alternatively, the environment may serve as a visual cue for performance or improvisation. Examples of this can be found in the works of Earle Brown. In his work *December 1952*, Brown used an algorithmic process to create a series of lines on a page which would serve as a score. Rather than compose linearly, Brown chose to compose spatially. Considering the area of the page as a grid, Brown used a random sampling table to determine the position, length, and thickness of the horizontal and vertical lines drawn on the page [8].

In this way, Brown developed a compositional method in which the space on the page literally determines the form of the music. Rather than consider the sound specifically, Brown only considered the visual/spatial elements of position, thickness, and direction of lines in composing the work. One reason for this decision is that Brown did not intend the piece to be performed from left-to-right, and therefore, it did not need to be composed in that manner [8]. Instead, the score of *December 1952* can be read in any direction, adding a further formal spatial component to the piece.

Other works of Brown employ a similar method of "spatial composition," including the octophonic tape work *Octet I* (1953). In this work, Brown used essentially the same method for composition as *December 1952*, using random numbers to determine the physical location of an event on the score, but instead of drawing lines on a page of a fixed length he placed splices of tape along a time continuum, using algorithmic processes to determine the timing of the event (the horizontal axis of the score) and the playback channel (the vertical axis of the score) [9].

4.1.1 Static Approach to Virtual Space in Zebra

The lead author's composition *Zebra* (2015) serves as an example of a musical composition for a static virtual space. This work aims to explore serendipitous and gradually transforming spatial textures of sound that can be obtained, organized, and presented in high-fidelity spatial sound for an audience, as the performer navigates through the virtual space shown in Figure 1.

Figure 1. A screenshot from *Zebra*, depicting the spheres that determine the spatial locations of the audio sources.

Zebra primarily consists of an arrangement/realization of a MIDI file released by composer Daniel Lopatin (a.k.a. Oneohtrix Point Never). In a similar approach to Brown's *Octet I*, sounds are placed in within a composed space. The sound itself is linear and pre-composed, but the virtual-physical environment in which the sound exists is generative and navigable.

The MIDI file (somewhat altered) is played back in musical time, driving a polyphonic synthesizer. The MIDI score represents a series of chords, and the individual notes of each chord are distributed so as to emanate from different objects within the virtual space. In this case, the virtual space is an environment created using the game engine Unity (`https://unity3d.com/`) and the objects are simple spheres with lights. These spheres are positioned randomly for each performance, so the layout is always different, and the notes are distributed to the spheres based on voice number in the polyphonic synthesizer (using Max's `poly~` object).

During performance, the performer navigates the virtual space wearing a VR head-mounted display, while the audience watches on a screen from the first-person perspective of the performer (similar to Lanier's *The Sound of One Hand*). Additionally, the sounds are spatialized so as to seem to emanate from their respective locations in virtual space. This spatialization is achieved using a recently developed Max extension called the Multi-source Ambisonic Spatialization Interface (MASI) (`http://zberkowitz.github.io/MASI/`).[1] A full screencast performance can be viewed at `https://vimeo.com/144070139`.

This static approach provides an interesting way to realize and explore musical content in new ways. It also gives the performer a certain agency to shape the formal content of the work. However, while this approach can maximize the performer's sense of agency, it does not fully engage the potential of the virtual space. In other words, the virtual world itself lacks agency. This issue is revisited below through a more dynamic approach to virtual world composition.

4.2 Dynamic Approaches

In contrast to using a static environment to navigate predetermined sonic content, one might consider instead using a dynamically changing environment to determine, change, and affect the sonic content. This approach could perhaps be more engaging in some contexts, and it can result in stronger conceptual ties between audio and visual components.

This is the case in Claude Cadoz's composition *pico..TERA* (2001), which uses a slowly oscillating (e.g. in a 1-6Hz range) mass-spring network to create rhythms by striking virtual percussion instruments. The particulars of the acoustic vibrations in the virtual percussion instruments affect the virtual mass-spring network, causing it to play complex rhythmic patterns, which are intriguing to listen to [11, 12]. In a sense, Claude Cadoz is using a virtual environment to generate a score, but the virtual environment is one-dimensional, so it is not really navigable.

Again, the work of Earle Brown serves as a compelling and historic example of dynamic composition using space. Brown was fascinated and inspired by the work of Alexander Calder. Calder was a sculptor known for his kinetic, hanging mobiles that helped to redefine modern sculpture. Brown desired to create music that was, like Calder's sculptures, re-configurable and therefore "mobile."

Similarly to what has been outlined in this paper, Brown considered two kinds of mobility:

> ...one the physical mobility of the score itself, and the other the conceptual mobility—which is to say the performer's mental approach to the piece—holding in mind the considerable number of different ways of moving, moving the mind around a fixed kind of graphic suggestion, or actually physically moving the score itself [8].

"Conceptual" mobility can be considered similar to the "static" approach defined here. The second kind of mobility—that in which the score itself is moving—can be considered the "dynamic" approach.

Later in his career, Brown would work together with Calder on *Calder Piece* (1966), for percussion quartet along with a mobile sculpture by Calder entitled "Chef d'orchestre" (conductor of the orchestra). The work, which was recently performed at the Tate Modern in November 2015, incorporates the Calder mobile in two ways. Firstly, at times the percussionists approach the mobile and use it as an instrument (the sculpture consists of metal plates that produce a gong-like sound when struck). Secondly, at other times the percussionists watch the movement of the mobile while playing other percussion instruments. In this configuration, the hanging parts of the mobile determine which parts of the score the percussionists should be playing. As Richard Bernas described the recent Tate Modern performance:

> ...movements of the sculpture are paralleled by the performers trajectories; ...improvised passages played on the sculpture italicize the more notated percussion solos; ...the integrity of the concept on a multiplicity of material and sonic levels creates continuity despite some surprises along the way. Though unfixed in some of its detail, the concept is clear and far from arbitrary. Brown and Calder demonstrate that flux, movement and uncertainty can indeed be positives [13].

The authors believe that 3D virtual space can be used to extend the concepts put forward in *Calder Piece* and related works. In addition, 3D virtual space can be used to resolve some of the practicalities involving a score that is itself a moving object. Finally, 3D virtual space can enable generation of complex spatial textures of sound, an idea explored below by one of the lead author's compositions.

4.2.1 Dynamic Approach to Virtual Space in Calder Song

Calder Song by the lead author is an example composition that utilizes a dynamic virtual space. The work is a variation on the idea of Brown's *Calder Piece* but with a different aesthetic approach.

Like *Zebra*, *Calder Song* employs Unity, Max, and MASI to create a 3D audiovisual space with realistic sound source locations that the performer can navigate among from a first-person perspective. However, *Calder Song* has moving parts in the form of Calder-esque virtual sculptures. Each of these sculptures demonstrates a different musical interaction. These interactions are more simple and direct than those in Brown's work, valuing a less improvisational aesthetic than Brown. A screencast performance of the piece can be viewed at `https://vimeo.com/163116373`.

Figure 2 shows an example of one of these virtual sculptures. The triangular hanging pieces in this sculpture move as though their connecting wires were attached to motors. Using the physics available in the Unity game engine, it is possible to build sculptures such as this that may be used to affect musical and artistic form through physics, as in Calder's mobiles and Brown's musical interpretations of them. Through the use of virtual space, it is much easier to realize Brown's idea for a moving, sculptural score.

As the mobile in Figure 2 turns, the hanging triangles generate notes when they become vertically aligned with other hanging triangles. The notes each triangle plays are determined by which other triangles they are vertically aligned with, and the sound emanates from the location of the triangle. In this way, the sculpture generates a tapestry of sounds that continually vary their rhythms and reconfigure themselves spatially. The balance and speed

[1] MASI, a software currently under development by the authors, is a series of patchers for Cycling '74's Max that provide a simplified interface for the realistic spatial positioning of sound sources in a virtual 3D environment through ambisonic panning and virtual acoustics [10]. MASI does not provide a graphical panning interface itself, but instead connects to other user-created graphical interfaces through Open Sound Control (OSC) communication. MASI is primarily intended to be used in conjunction with 3D game-like virtual world environments/interfaces. Scripts are provided to connect MASI with the Unity game engine.

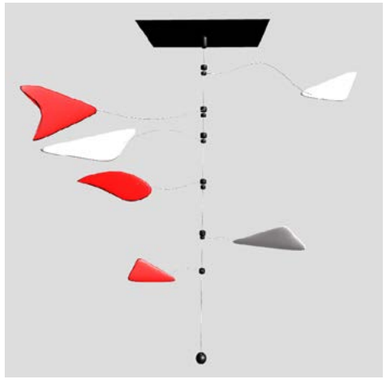


Figure 2. A screenshot from *Calder Song*, depicting one of the music-generating virtual mobiles.

of the virtual mobile determine the musical trajectory of the "part"to which the sculpture is assigned, as part of the greater "song."

In summary, in *Calder Song*, multiple virtual sculptures work together to control the sonic content. The interactions between virtual sculpture and sound can vary widely, and this virtual sculpture "garden" environment has the potential to further explore Brown's concept of musical form defined, manipulated, and controlled by physical processes.

5. CONCLUSIONS

Throughout his career, Earle Brown experimented with conceptually mobile scores, in which the performer had the agency to move about a fixed score space, and scores that were actually physically mobile. In *Calder Piece*, he realized a truly physically mobile score in which performers reacted to and interacted with a kinetic sculpture that served to alter the form and direction of the music.

Virtual reality provides a new frontier for composers to work with 3D space as a score or as a controller of musical form. Tools such as Unity and MASI aim to make this process easier than ever. By applying Brown's ideas to virtual worlds, an area of compositional research related to score, form, and space can be explored and expanded. Considering dynamic versus static approaches, composers can actively manage the creation of agency for performer, score, and/or audience. It seems that virtual reality is indeed an excellent medium for developing, re-examining, and re-imagining mobile form.

Acknowledgments

The authors would like thank the LSU School of Music, Center for Computation and Technology, and Experimental Music and Digital Media program for continuing support on this research.

6. REFERENCES

[1] B. G. Tyranny, "Out to the Stars, Into the Heart: Spatial Movement in Recent and Earlier Music," *NewMusicBox: The Web Magazine*, January 1 2003. [Online]. Available: http://www.newmusicbox.org/page.nmbx?id=45tp00

[2] V. Lombardo, A. Arghinenti, F. Nunnari, A. Valle, H. H. Vogel, J. Fitch, R. Dobson, J. Padget, K. Tazelaar, S. Weinzierl, S. Benser, S. Kersten, R. Starosolski, W. Borczyk, W. Pytlik, and S. Niedbaa, "The Virtual Electronic Poem (VEP) Project," in *Free Sound*, ser. Proceedings of the International Computer Music Association. San Francisco: International Computer Music Association, 2005, pp. 451–4.

[3] V. Lombardo, A. Valle, J. Fitch, K. Tazelaar, S. Wenzierl, and W. Borczyk, "A Virtual-Reality Reconstruction of Poeme Electronique Based on Philological Research," *Computer Music Journal*, vol. 33, no. 2, pp. 24–47, 2009.

[4] J. Lanier, "The Sound of One Hand," *Whole Earth Review*, no. 79, pp. 30–4, 1993.

[5] C. Cruz-Neira, D. J. Sandin, T. A. DeFantl, R. V. Kenyon, and J. C. Hart, "The Cave: Audio Visual Experience Automatic Virtual Environment," *Communications of the ACM*, vol. 35, no. 6, pp. 64–72, 1992.

[6] R. Hamilton, "The Procedural Sounds and Music of ECHO::Canyon," in *Music Technology Meets Philosophy: From Digital Echos to Virtual Ethos*, ser. Proceedings of the International Computer Music Association. San Francisco: International Computer Music Association, 2014, vol. 1, pp. 449–55.

[7] ——, "Sonifying Game-Space Choreographies with UDKOSC," in *Proceedings of the 13th International Conference on New Interfaces for Musical Expression*. Daejeon, Korea: New Interfaces for Musical Expression, 2013, pp. 446–9.

[8] E. Brown, "On December 1952," *American Music*, vol. 26, no. 1, pp. 1–12, 2008.

[9] V. Straebel, "Interdependence of Composition and Technology in Earle Browns Tape Compositions Octet I/II (1953/54)," paper presented at Beyond Notation: An Earle Brown Symposium, Northeastern University, Boston, January 18-19, 2013.

[10] J. Schacher, "Seven Years of ICST Ambisonics Tools for MaxMSP - A Brief Report," in *Proc. the 2nd International Symposium on Ambisonics and Spherical Acoustics*, Paris, France, May 6-7 2010.

[11] C. Cadoz, "The Physical Model as Metaphor for Musical Creation. pico..TERA, a Piece Entirely Generated by a Physical Model," in *Proceedings of the International Computer Music Conference*, Göteborg, Sweden, 2002.

[12] ——, "Supra-Instrumental Interactions and Gestures," *Journal of New Music Research*, vol. 38, no. 3, pp. 215–230, September 2009.

[13] R. Bernas, "Flux, Movement, and Uncertainty," *Journal of the Institute of Composing*, no. 7, 2016. [Online]. Available: http://www.instituteofcomposing.org/journal/issue-7/flux-movement-and-uncertainty/

Description of Chord Progressions by Minimal Transport Graphs Using the System & Contrast Model

Corentin Louboutin
Université de Rennes 1 / IRISA
corentin.louboutin@irisa.fr

Frédéric Bimbot
CNRS / IRISA
frederic.bimbot@irisa.fr

ABSTRACT

In this paper, we model relations between chords by minimal transport *and we investigate different types of relations within chord sequences. For this purpose, we use the "System & Contrast" (S&C) model [1, 2], designed for the description of music segments, to infer non-sequential structures called* chord progression graphs *(CPG). Minimal transport is defined as the shortest displacement of notes, in semitones, between a pair of chords. The paper presents three algorithms to find CPGs for chords sequences: one is sequential, and two others are based on the S&C model. The three methods are compared using the perplexity as an efficiency measure. The experiments on a corpus of* 45 *segments taken from songs of multiple genres, indicate that optimization processes based on the S&C model outperform the sequential model with a decrease in perplexity over* 1.0.

1. INTRODUCTION

One of the topics of major interest in Music Information Retrieval (MIR) is to understand how elements are related to one another in a music piece. For this purpose, some studies use principles from formal language theories [3, 4, 5], some others formalize notions from conventional musicology [6, 7] and another branch in music information retrieval is mainly based on probabilistic models [8, 9].

Recently Bimbot et al. designed the "System & Contrast" model [1, 2] to describe music at the scale of phrases and *sections*, i.e. segments of 12 to 25 seconds, typically from songs. The S&C model is a multidimensional model which can be applied to melody, harmony, rhythm or any other musical dimension. The S&C model is based on the idea that relations between musical elements are not essentially sequential and that they can be infered on the basis of an economy principle. We focus here on the application of this model to the description of chord progression structures.

The study presented in this paper is based on the notion of *minimal transport* which is used to model the relation between two chords. It is defined as the set of connections between the notes of the two chords such that the sum of intervals (in semitones) resulting from the connections is minimal. As such, the notion of minimal transport can be seen as a computational approximation of "voice leading" as described by Cohn [10] or Tymoczko [11, 12]. However, minimal transport is here extended to also infer non-sequential structures which is a way to describe how chords are related, to one another, while relaxing the sequentiality hypothesis.

It was observed in Deruty et al. [13] that it is possible to create a multi-scale segment structure using the S&C model at different scales simultaneously. The present paper investigates the computational potential of this hypothesis for minimal transport graph search.

In Section 2.3, we define the notion of chord progression graph (CPG) and minimal transport graph (MTG), and we briefly recall the square form of the System & Contrast model. We then describe in Section 3 three optimization algorithms, one sequential and two based on the S&C model, to compute a minimal transport chord sequence. Finally, in Section 4, we present an experimental comparison of these three optimization methods, in terms of perplexity.

2. KEY CONCEPTS

2.1 Definitions

A *chord sequence* can be defined as the *in extenso* representation of all chords observed in a segment at specific metric positions and ordered by time. A chord is itself represented by the set of pitch classes (pc) of each note composing it.

A *chord progression graph* (CPG) is a pair (S, M) where S is a sequence of chords and M is the *model structure* of relations between the chords, that is the set of links between them. Two kinds of CPGs are considered in this paper:

- *sequential* CPGs which are based on the sequential description of the chord sequence. For these graphs, each link defines a relation between a chord and the chord appearing just after, in the chord sequence.

- *systemic* CPGs, based on the S&C model described in subsection 2.3 for which relations between chords are causal but not necessarily sequential.

While for sequential CPGs the *antecedent* of a chord is its immediate predecessor, it can be some other previous chord for systemic CPGs. In both cases we make the hypothesis that a given chord, S_i, depends only on one antecedent, $\Phi(S_i)$, itself of the chord type. Using a probabilist point of view, we can use Φ to define an approxima-

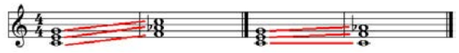


Figure 1. Example of transports between the two chords C and Fm. The first is the $\{(0,5),(4,8),(7,0)\}$ and the second is $\{(0,0),(4,5),(7,8)\}$

tion of $P(S_i|S_{i-1}\dots S_0)$:

$$P(S_i|S_{i-1}\dots S_0) \approx P(S_i|\Phi_M(S_i)) \quad (1)$$

where M denotes the model structure of the CPG.

For the sequential CPG, $\Phi_{Seq}(S_i) = S_{i-1}$, and this is equivalent to a first order Markov approximation. When Φ_M is deterministic, the CPG (S, M) can be denoted by the pair (S, Φ_M).

2.2 Minimal Transport

A chord P is represented by a set of m_p pitch classes p_i: $P = (p_i)_{0\leq i\leq m_p}$. A *transport* between P and an other chord $Q = (q_j)_{0\leq j\leq m_q}$ is a set :

$$T = \{(p_k, q_k)|p, q \in [\![0; 11]\!], k \in [\![0; n]\!]\} \quad (2)$$

where n is the number of connections or voices. Indeed a transport can be seen as a way to associate voices to notes in the two chords. We focus here on complete transports, i.e. each note is associated to at least one voice. Examples of such transports are given on Figure 1.

The optimality of a transport between two chords is defined by the *taxicab norm* or *smoothness* [14]. That is, for a transport T:

$$|T| = \sum_{(p,q)\in T} |d(p, q)| \quad (3)$$

where

$$d(p, q) = ((q - p + 5) \pmod{12}) - 5 \quad (4)$$

The term $d(p, q)$ is the shortest displacement in semitones from pitch class p to pitch class q (with $d(p, q) \in [\![-5; 6]\!]$).

In Figure 1 the second transport is minimal. A *minimal transport graph* (MTP) is an instantiation of a CPG (S, M) where all transports associated with M are fixed and their sum is minimal.

2.3 System & Contrast Model

The System & Contrast (S&C) model [2] is a (meta-)model of musical data based on the hypothesis that the relations between musical elements in a segment are not necessarily sequential. Initially designed for the description of phrase structure for annotation purposes [1], the S&C model has been further formalized as a generalization and an extension of Narmour's implication-realisation model. Its applications to various music genres for multidimensional and multiscale description has been explored in [13]. Our aim, here is to give a computational elaboration of this model.

The principles of the S&C model is that relations between elements in a musical segment create a *system* of matricial expectations which can be more or less strongly denied by the last element called *contrast*. The first element of the system is called *primer* and plays a particular role in the construction of the expectation system. The contrast acts as a closure to the segment. In this paper, we focus on *square* systems, i.e. systems of four elements.

2.3.1 Formalization

A sequence of four elements $(x_i)_{0\leq i\leq 3}$ can be arranged as a square matrix:

$$X = \begin{bmatrix} x_0 & x_1 \\ x_2 & x_3 \end{bmatrix} \quad (5)$$

Assuming two relations f and g between the primer x_0 and its neighbors in X, we have:

$$x_1 = f(x_0) \quad \text{and} \quad x_2 = g(x_0)$$

Note that these two relations may apply only to a subset of the properties characterizing the elements of the system.

The S&C model envisions the fourth element x_3 in relation to a *virtual* projected element $\hat{x_3}$ which would result from the combination of f and g: The disparity between $\hat{x_3}$ and the actual (observed) x_3 is modeled by a *contrast* function γ:

$$\hat{x_3} = f(g(x_0)) \quad (6)$$

$$x_3 = \gamma(\hat{x_3}) \quad (7)$$

The description of a S&C is the quadruplet (x_0, f, g, γ) which can be used as a compact representation of the segment. It can be viewed as a minimal description in the sense of the Kolmogorov complexity [15] in line with several other works in MIR [16, 17, 18].

For a chord sequence $(S_i)_{0\leq i\leq 3}$ modeled as a S&C, the antecedent function $\Phi_{S\&C}$ (see Eq. 1) is defined as follows:

$$\Phi_{S\&C} : \begin{matrix} S_1 \longmapsto S_0 \\ S_2 \longmapsto S_0 \\ S_3 \longmapsto \hat{S_3} \end{matrix} \quad (8)$$

Under the minimal transport approach, f, g and γ are complete transports.

2.3.2 Multiscale S&C

Musical phrases and sections generally contain time varying chord information which can be sampled at specific intervals, for instance downbeats. In this work, chord progressions are assumed to be composed of 16 elements, for instance:

$$Cm\ Cm\ Cm\ Bb\ Ab\ Ab\ Ab\ Gm\ F\ F\ F\ F\ Cm\ Cm\ Bb\ Bb$$

The S&C model can be used to model such sequences by extending it to a multiscale framework [13].

A multiscale CPG is a structure that combines elementary sub-CPGs built from square S&Cs. Figure 2 represents a view of the above chord sequence explained on several scales simultaneously as a hypercube or *tesseract*. Bolded chords are contrastive elements. In the first system $[Cm, Cm, Cm, Bb]$, Bb contrasts with the expectation: $Cm + Cm + Cm \rightarrow Cm$. In the second group, Gm denies $Ab{+}Ab{+}Ab \rightarrow Ab$. The last F is a non-contrastive chord in the $[F, F, F, F]$ group. Ultimately, the sequence concludes by $Cm + Cm + Bb \rightarrow Bb$.

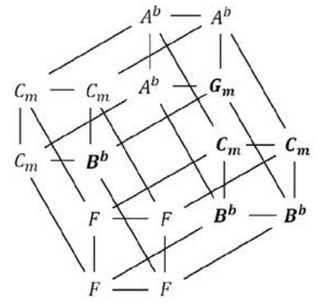


Figure 2. Tesseractic representation of the chord sequence of the chorus of *Master Blaster* by Stevie Wonder.

But many other systems can also be considered. For instance chords number $[0, 1, 4, 5]$ form a non-contrastive subsystem $[Cm, Cm, Ab, Ab]$, while chords $[8, 10, 12, 14]$ form a contrastive sub-system, $[F, F, Cm, Bb]$, etc. In fact, any quadruplet of adjacent vertices forming a square in the tesseract can be considered as a S&C. This results in a graph of implications which describes the chord sequence in a multiscale fashion.

3. APPLICATION TO CHORD PROGRESSION ANALYSIS

Finding the Minimum Transport Graph (MTG) on a chord sequence is an optimization problem. It consists in finding the global transport graph whose transport cost is minimal. In this section, we present three structure models designed for 16-chord sequences, and the corresponding optimization algorithms: namely the sequential model (Seq), the bi-scale model ($SysP$) and the dynamic scale model ($SysDyn$).

Each optimization process described below explores the space of all transport graphs corresponding to a CPG and chooses the solution with the minimal global cost.

3.1 Sequential Model

The sequential model corresponds to the conventional point of view where each chord is related to its direct predecessor, i.e. $\Phi_{Seq}(S_i) = S_{i-1}$.

As the number of possible transport graphs grows very fast ($\mathcal{O}(n!^l)$) with the length of the chord sequence l and the number of voices n (as defined in Equation 2), we divide the transport graph optimization in local optimizations of complexity $\mathcal{O}(n!^{\frac{l}{4}+1})$ on four sub-graphs. The first optimization is the search of the minimal transport graph corresponding to the CPG $([S_i]_{0\leq i\leq 3}, \Phi_{Seq})$. Then, the algorithm builds a second CPG using the last chord of the previous optimization and the four next chords of S, that is $([S_i]_{3\leq i\leq 7}, \Phi_{Seq})$, and searches for the corresponding MTP. This step is iterated on sequences $[S_i]_{7\leq i\leq 11}$ and $[S_i]_{11\leq i\leq 15}$.

Figure 3 represents the global model structure of the sequential model. Each transport links a chord with the next chord in the sequence and the graph is optimized by groups of four or five chords.

Figure 3. Representation of the transport links of the sequential model. Each colored sub-graph represents one optimization. Numbers are the chord indexes in the initial sequence.

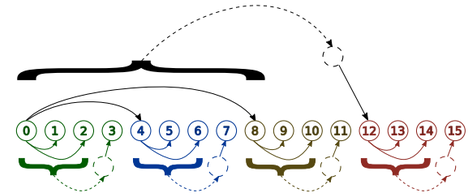


Figure 4. Representation of the transport links of the bi-scale model. In black, the upper scale system, while the colored systems are the lower scale systems. Numbers are the chord indexes in the initial sequence.

3.2 Static Bi-Scale Model

The second structure model is based on a multiscale vision of the S&C model as described in Section 2.3.2. A sequence of 16 chords can be structured in a S&C of four disjoint nested sub-CPGs with a S&C structure, in other words, a S&C of four S&Cs. Under this approach:

- An upper scale CPG models the systemic relations between the first elements of the four lower scale CPGs: $([S_0, S_4, S_8, S_{12}], \Phi_{S\&C})$.
- The four lower scale CPGs describe the structure of four disjoint parts of the segment: $([S_{4*i+j}]_{0\leq j\leq 3}, \Phi_{S\&C})_{0\leq i\leq 3}$.

The global bi-scale model is represented on Figure 4. The function of the upper scale CPG is to ensure the global coherence of the description.

As for Seq model, each optimization has to be computed separately to reach a reasonable computing time ($\mathcal{O}(n!^{\frac{l}{4}})$).

As relations in a S&C are matricial, they become tensorial at the multiscale level (see Figure 2), and it is therefore interesting to consider *permutations* of the initial sequence such that each of the five S&Cs corresponds to a square in the tesseract and each point of the tesseract appears only in one lower scale CPG (see Section 2.3.2).

Moreover, to ensure that each CPG can be described using a S&C model structure, chord indexes of each CPG have to correspond to a quadruplet forming an adjacent square in the tesseract view (see Figure 2). There are only 36 possibilities of such permutations which respect local causality inside each CPG.

As a system of 4 elements, $abcd$, is equivalent to its dual, $acbd$, due to the fact that both b and c are related commutatively to a in the MTG approach. Using this equivalence on the upper CPG, it is possible to reduce the 36 permutations to 30 equivalence classes [1] . For example, the permutation $[0, 1, 2, 3, 8, 9, 10, 11, 4, 5, 6, 7, 12, 13, 14, 15]$ is equivalent to the one represented on Figure 4. A bi-scale model structure associated with a permutation number x

[1] The list of permutations is given in [19]

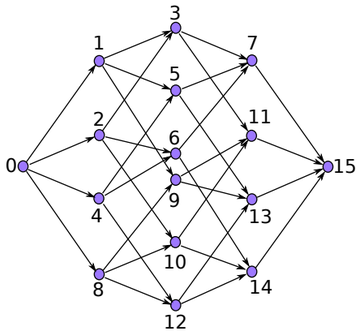


Figure 5. Projection of the tesseract where each chord of a same column has the same contrastive function. Numbers are the chord indexes in the initial sequence.

is denoted as $SysPx$ ($SysWPx$ in the case where $\hat{x_3}$ is replaced by x_0).

3.3 Dynamic Scale Model

3.3.1 Principle

This third structure model, denoted as $SysDyn$, is also based on the S&C model and the tesseractic representation of the chord sequence. But, while the arrangement of nested systems is fixed by the permutation in the bi-scale model, the dynamic model considers a wider range of combinations. Figure 5 represents the tesseract in a way such that, each column aligns the chords having the same contrastive depth in the sequence (i.e. they are contrastive elements for a same number of systems). The first column contains only the primer, the second column contains the secondary primers $(1, 2, 4, 8)$ (which are not contrastive elements of any system), then on the third column, the contrastive elements of only one system $(3, 5, 6, 9, 10, 12)$. Then, elements 7, 11, 13 and 14 can act as contrastive elements of three systems and the final element (15) is potentially contrastive in six systems.

The principle of the dynamic method is to optimize on the fly the sub-CPGs which contribute to the MTG of the overall chord progression. For instance, chord 11 is hypothesized as the contrast of sub-CPGs:

- $([S_1, S_3, S_9, S_{11}], \Phi_{S\&C})$
- $([S_2, S_3, S_{10}, S_{11}], \Phi_{S\&C})$
- $([S_8, S_9, S_{10}, S_{11}], \Phi_{S\&C})$

Among these three possibilities, the one yielding the minimal transport graph is selected dynamically as the local structure within the global description. Therefore, this requires a two level optimization process: one for the search of the best sub-CPG that "explains" a contrastive element and one for the transport graph of each sub-CPG.

3.3.2 Handling optimization conflicts

To prevent optimization conflicts when two different CPGs contain the same relation (e.g. (S_0, S_1) in $[S_0, S_1, S_2, S_3]$ and $[S_0, S_1, S_8, S_9]$), each transport is fixed at the optimization of the first CPG in which it appears. It implies that when optimizing the CPG, $([S_0, S_1, S_8, S_9], \Phi_{S\&C})$, the transport considered for (S_0, S_1) is fixed and is the one considered in the minimal transport graph associated with the CPG: $([S_0, S_1, S_2, S_3], \Phi_{S\&C})$.

Moreover this constraint also applies to the voices associated with each note. If a former optimization step has determined the voices relating two chords, the transport between these two chords is kept fixed for the forthcoming sub-CPGs optimizations. For example, once the optimization on CPG $([S_0, S_1, S_8, S_9], \Phi_{S\&C})$ has been achieved, the voices associated with pitch classes of S_1 and S_9 fixes the transport between these two chords for later optimizing the CPG $([S_1, S_5, S_9, S_{13}], \Phi_{S\&C})$. In the current implementation of the algorithm, the CPG optimizations are carried out in ascending order of the index of the contrastive element of the CPGs, which preserves causality.

4. EXPERIMENTS

4.1 Data

In this section, we present experimental results on the behaviour of the proposed models on a dataset of 45 structural sections from a variety of songs, reduced to 16 (downbeat sychronous) chord sequences, including artists such as Miley Cyrus, Edith Piaf, Abba, Pink Floyd, Django Reinhardt, Eric Clapton, Rihanna, etc [2] .

4.2 Evaluation

4.2.1 NLL Score

As there exists no ground truth as of the actual structure of the chord sequences, we compare the different models with regards to their ability to predict the entire chord sequence in the CPG framework. This is done by calculating a *perplexity* [20] for each model derived from the *negative log-likelihood*, denoted as NLL_M.

The NLL_M of a transport graph is defined as the arithmetic mean of the NLL_M of each voice inferred by the transport graph. Let $X = (x)_{0 \leq i \leq n-1}$ be the sequence of pitch classes of a "voice", considering the first-order approximation defined in Section 2.1, the NLL_M associated with a CPG M, is defined as:

$$NLL_M(X) = -\frac{\log p(x_0) + \sum_{d \in D_M} \log p(d)}{|D_M| + 1} \quad (9)$$

where D_M is the set of pitch class displacements in semitones in the voice considering the CPG structure model, $|D_M|$ is the size of the set [3], and $p(d)$ is the estimated probability of the displacement d.

4.2.2 Probability Estimation

In this work, $p(d)$ is estimated as:

$$p(d) = \frac{1 + N(d, C_M)}{12 + \sum_{z=-5}^{6} N(z, C_M)} \quad (10)$$

where C_M is the description of the training corpus with the model M (using a leave-one-out cross-validation strategy),

[2] The full list of chord sequences is presented in [19].

[3] For $SysDyn$, if a displacement is used in two sub-CPGs, the displacement is counted twice for the likelihood.

	$\hat{x_3} = f(g(x_0))$	$\hat{x_3} = x_0$
Seq	3.84	
$BestSysP$	2.73	2.95
$SysP8$	3.17	3.64
$SysDyn$	2.80	

Table 1. Average perplexity obtained by the different models.

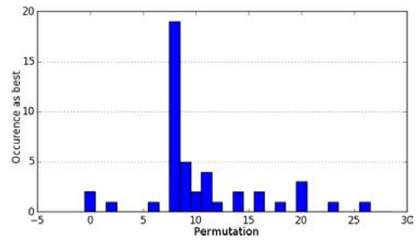


Figure 7. Histogram of the top ranking permutations, in terms of perplexity, across the 45 chord sequences.

and $N(d, C_M)$ the number of occurences of displacements d observed in C_M. $p(d)$ is an estimation of $P(x|\Phi_M(x))$ where $d(\Phi_M(x), x) = d$.

We hypothetize the *a priori* uniformity in the distribution of the initial notes and therefore estimate $p(x_0) = \frac{1}{12}$ which preserve the comparability between the models.

4.2.3 Perplexity

We convert a NLL into a perplexity value defined as:

$$PP_M(S) = 2^{NLL_M(S)} \quad (11)$$

which can be interpreted as the average probabilistic branching factor between successive notes in the graph.

4.3 Results

Figure 6 depicts a comparison between Seq, $BestSysP$, and $SysDyn$ models for each of the 45 chord sequences where $BestSysP$ is defined as the optimal permutation of the $SysP$ configuration for each song individually.

Table 1 summarizes the results with the three types of models. While the sequential model (Seq) provides a perplexity of 3.84, it is clearly outperformed by both the bi-scale model and the dynamic model, 2.73 and 2.80 respectively, i.e. more than 1.0 perplexity difference.

It is worth noting from Figure 7 that permutation 8 (represented on Figure 8) is the optimal permutation for 19 songs out of 45 (i.e. 42%). An explanation of the success of this permutation can be that it considers implicitly three types of scale relations: short, medium and long. The upper scale optimization maximizes the coherence of the first half of the chord sequence, while lower scale optimizations combine local and distant relations.

In the context of the bi-scale model, the role of the virtual element, $\hat{x_3}$ in $SysPx$ has been investigated experimentally by substituting it with the primer x_0 in the CPGs, in order to compare both of them as predictors of x_3. The second column of Table 1 shows a clear advantage of the virtual element which comforts the idea of its implicative role in the S&C model. However, there are 5 chord sequences for which x_0 is significantly better than $\hat{x_3}$. This may happen when the last element falls back on the primer

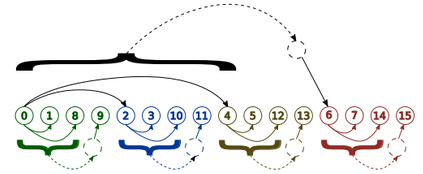


Figure 8. CPG of the bi-scale model, using permutation 8: $[0, 1, 8, 9, 2, 3, 10, 11, 4, 5, 12, 13, 6, 7, 14, 15]$.

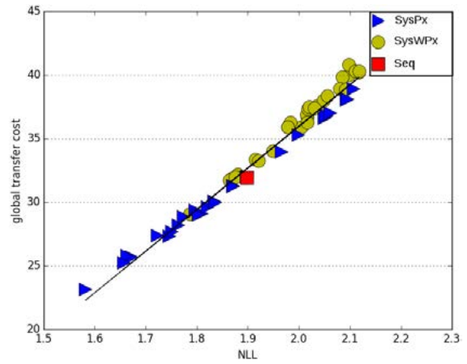


Figure 9. Graph of average transport cost, function of average NLL on 45 sequences corpus for Seq, $SysP0 \ldots 29$ and $SysWP0 \ldots 29$.

or if the virtual element contains pitch classes which do not belong to the "tonality" of the segment.

As optimizing the transport cost between chords minimizes the average pitch class displacement, there are only few intervals capturing most of the NLL. This raises the idea, as Figure 9 shows, that there is a correlation (0.990) between the global transport cost and the NLL. This may indicate that the distribution of the displacement distances is somehow exponentially decreasing. It would therefore be interesting to investigate how replacing the trained probability estimations by a Laplacian law would affect the results.

Finally, $SysDyn$ happens to perform equivalently well to $BestSysP$ but with a much faster computation time. The optimal model structure can be traced back *a posteriori*. Interestingly, a chord that is contrastive in a CPG can then be used in a new CPG to build the expectation for a subsequent contrastive "surprise". In a sense it can be seen as a similar notion to that of "resolution" in conventional musicology [21, 10]—with the difference that, here, the resolution is realised from a virtual chord.

In summary, this first set of results shows that considering non-sequential relations between chords seem relevant to provide an efficient description of chord progressions.

5. CONCLUSIONS AND PERSPECTIVES

The approach presented in this paper is based on minimal transport to model relations between chords. Three optimization algorithms have been presented and tested on a corpus of 45 sequences of 16 chords using perplexity as an efficiency measure. The two methods based on the S&C model substantially outperform the sequential approach.

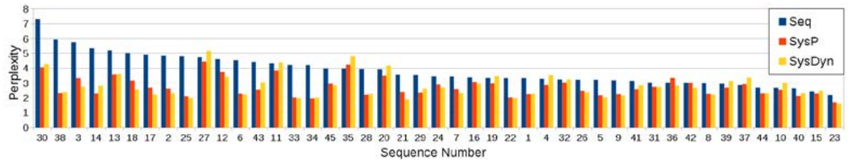


Figure 6. Perplexity obtained for each of the 45 chord sequences by: Seq (sequential model), $BestSysP$ (optimal bi-scale permutation for each song) and $SysDyn$ (dynamic model).

These results constitute a strong incentive to further consider the use of the S&C model in MIR.

The S&C model could also prove to be useful in musicology: in particular, the virtual element considered by the S&C model seems to play a relevant role. It may have a similar function to that of the augmented triad in Cohn's theory [10], that is, a passage chord which can be "invisible" in the observed sequence. Future studies could investigate how the definition of the virtual element affects the MTG optimization and how to constraint transports to comply with musicological rules.

Furthermore, we focused here only on the chord dimension of music, but the System & Contrast model can handle other dimensions such as melody, rhythm, etc, which will be a subject for future investigations.

Acknowledgments

This work has greatly benefited from initial scientific investigations carried out with Anwaya Aras, during her internship at IRISA, in 2013.

6. REFERENCES

[1] F. Bimbot, E. Deruty, G. Sargent, and E. Vincent, "Semiotic structure labeling of music pieces: concepts, methods and annotation conventions," in *Proc. ISMIR*, 2012.

[2] F. Bimbot, E. Deruty, G. Sargent, and E. Vincent, "System & Contrast : A Polymorphous Model of the Inner Organization of Structural Segments within Music Pieces," *Music Perception*, vol. 33, pp. 631–661, June 2016. Former version published in 2012 as Research Report IRISA PI-1999, hal-01188244.

[3] B. de Haas, J. P. Magalhaes, R. C. Veltkamp, and F. Wiering, "Harmtrace: Improving harmonic similarity estimation using functional harmony analysis.," in *Proc. ISMIR*, pp. 67–72, 2011.

[4] W. B. De Haas, M. Rohrmeier, R. C. Veltkamp, and F. Wiering, "Modeling harmonic similarity using a generative grammar of tonal harmony," *Proc. ISMIR*, 2009.

[5] M. Rohrmeier, "A generative grammar approach to diatonic harmonic structure," in *Proceedings of the 4th Sound and Music Computing Conference*, pp. 97–100, 2007.

[6] M. Giraud, R. Groult, and F. Levé, "Computational analysis of musical form," in *Computational Music Analysis*, pp. 113–136, Springer, 2016.

[7] M. Giraud, R. Groult, and F. Levé, "Subject and countersubject detection for analysis of the well-tempered clavier fugues," in *From Sounds to Music and Emotions*, pp. 422–438, Springer, 2013.

[8] R. Scholz, E. Vincent, and F. Bimbot, "Robust modeling of musical chord sequences using probabilistic n-grams," in *ICASSP 2009*, pp. 53–56, IEEE, 2009.

[9] D. Conklin, "Multiple viewpoint systems for music classification," *Journal of New Music Research*, vol. 42, no. 1, pp. 19–26, 2013.

[10] R. Cohn, *Audacious Euphony: Chromatic Harmony and the Triad's Second Nature*. Oxford University Press, 2011.

[11] D. Tymoczko, "The geometry of musical chords," *Science*, vol. 313, no. 5783, pp. 72–74, 2006.

[12] D. Tymoczko, "Scale theory, serial theory and voice leading," *Music Analysis*, vol. 27, no. 1, pp. 1–49, 2008.

[13] E. Deruty, F. Bimbot, and B. Van Wymeersch, "Methodological and musicological investigation of the System & Contrast model for musical form description," Research Report RR-8510, INRIA, 2013. hal-00965914.

[14] J. N. Straus, "Uniformity, balance, and smoothness in atonal voice leading," *Music Theory Spectrum*, vol. 25, no. 2, pp. 305–352, 2003.

[15] P. M. Vitányi and M. Li, "Minimum description length induction, Bayesianism, and Kolmogorov complexity," *IEEE Trans. Information Theory*, vol. 46, no. 2, pp. 446–464, 2000.

[16] P. Mavromatis, "Minimum description length modelling of musical structure," *Journal of Mathematics and Music*, vol. 3, no. 3, pp. 117–136, 2009.

[17] D. Temperley, "Probabilistic models of melodic interval," *Music Perception*, vol. 32, no. 1, pp. 85–99, 2014.

[18] C. Louboutin and D. Meredith, "Using general-purpose compression algorithms for music analysis," *Journal of New Music Research*, 2016.

[19] C. Louboutin and F. Bimbot, "Tensorial Description of Chord Progressions - Complementary Scientific Material," IRISA, 2016. hal-01314493.

[20] P. F. Brown, V. J. D. Pietra, R. L. Mercer, S. A. D. Pietra, and J. C. Lai, "An estimate of an upper bound for the entropy of english," *Computational Linguistics*, vol. 18, no. 1, pp. 31–40, 1992.

[21] A. Forte, *Tonal harmony in concept and practice*. Holt, Rinehart and Winston, 1974.

Do Nested Dichotomies Help in Automatic Music Genre Classification? An Empirical Study

Tom Arjannikov
University of Victoria
tom.arjannikov@gmail.com

John Z. Zhang
University of Lethbridge
zhang@cs.uleth.ca

ABSTRACT

Dichotomy-based classification approaches are based on decomposing the class space of a multiclass task into a set of binary-class ones. While they have been shown to perform well in classification tasks in other application domains, in this work we investigate whether they could also help improve genre classification in music, a core task in Music Information Retrieval. In addition to comparing some of the existing binary-class decomposition approaches, we also propose and examine several new heuristics to build nested dichotomy trees. The intuition behind our heuristics is based on the observation that people find it easy to distinguish between certain classes and difficult between others. One of the proposed heuristics performs particularly well when compared to random selections from all possible balanced nested dichotomy trees. In our investigation, we use several base classifiers that are common in the literature and conduct a series of empirical experiments on two music datasets that are publicly available for benchmark purposes. Additionally, we examine some issues related to the dichotomy-based approaches in genre classification and report the results of our investigations.

1. INTRODUCTION

Music Information Retrieval (*MIR*) is a fast-growing interdisciplinary research area across information retrieval, computer science, musicology, psychology, etc. It focuses on managing large-volume music repositories, facilitating operations such as indexing, retrieval, storage, queries, etc. The driving force behind MIR comes from the recent technological advances, such as larger data storage, faster computer processing speed, etc., and the demanding need to tackle the ever growing amount of digitized music data [1].

Genre classification in music, i.e. categorizing music pieces into classes such that subsequent operations (mainly querying) could be easily conducted, is usually treated as one of the introductory steps toward high-level MIR tasks, including automatic tag annotation, recommendation, playlist generation, etc. While music genres are still largely regarded as ambiguous and subjective, musicians and listeners alike still use them to categorize music. Computational approaches are actively sought to automate the genre classification process [1].

The work to be presented in this paper follows this direction. [1] *Dichotomy* is a partitioning method in which an entirety is decomposed into two parts that are both: jointly (or collectively) exhaustive, i.e., a member only belongs to one part or the other, and mutually exclusive, i.e. no member belongs to both parts simultaneously. In our experience with music data in genre classification, when considering a group of music genres as an entirety, we often find that the classification accuracy degrades as the number of genres increases. Based on these observations, it would be interesting to find out whether separating music genres into subgroups could help improve the overall classification accuracy. We conjecture that dichotomy-based classification could be a potential way to do it.

Our efforts here are focused on finding new dichotomy-based approaches to music genre classification, which perhaps also could be useful in other application domains. In this work we use *content-based features* (not metadata), which are extracted directly from music and represent the different acoustic characteristics of the music sound [1]. In a larger setting, dichotomy-based classification is an ensemble learning approach that combines multiple classification models to solve a computational intelligence problem and aims to achieve better classification accuracy than the individual ones [3]. Ensemble learning has been gaining popularity in the non-trivial task of multiclass classification.

To our surprise, the results of our extensive experiments suggest that dichotomy-based approaches do not perform as well as expected in music genre classification. This is an interesting observation. We attempt to discuss and analyze this situation and hope that our investigations in this work would shed light on the future endeavors using dichotomy-based approaches to classification problems in music data.

2. PREVIOUS WORKS

Classification is the process of organizing objects into predefined classes. It is one of the core tasks in MIR. Music genres, emotions in music, music styles, instrument recognition, etc. are typical classification problems in MIR. Due to the ambiguity and subjectivity in the cognitive nature of music, classification is usually a hard task. Tzanetakis and Cook are among the first to work on this problem, specifically on labeling an unknown music piece with a correct genre name [4]. They show that this is a difficult problem even for humans and report that college students achieve no higher than 70% accuracy.

[1] We have reported the preliminary results of this work as a poster [2].

Meng and Shawe-Taylor [5] model short-time features in music data and study how they can be integrated into a Support Vector Machine (SVM) kernel. Li and Sleep [6] extend normalized information distance into kernel distance for SVM and demonstrate classification accuracy comparable to others. DeCoro et al. [7] use Bayesian Model to aid in hierarchical classification of music by aggregating the results of multiple independent classifiers and, thus, perform error correction and improve overall classification accuracy. Through empirical experiments, Anglade et al. [8] use decision tree for music genre classification by utilizing frequent chord sequences to induce context free definite clause grammars of music genres.

Silla *et al.* [9] perform genre classification on the same dataset as one of the two used in this paper, i.e. *Latin Music Database*. They use Naïve-Bayes, Decision Tree, etc., as the base classifier in a majority-vote ensembles. The best result in their work is based on the space- and time-decomposition of music data, which results in a set of classifiers, whose predictions are combined by a simple majority voting mechanism, i.e. for a new music piece, select its class as the one selected by the majority of the member classifiers in the ensemble. Sanden and Zhang [10] discuss a set of ensemble approaches and their application in genre classification, including maximization rules, minimization rules, etc. Due to the active research in music genre classification and for the sake of space, we mention only a few of the most relevant works here. For broader discussions, an interested reader is referred to Li *et al.* [1] and Strum [11].

Music genre classification naturally can be modeled as a multiclass problem. In machine learning, dichotomy-based approach is a statistical way to deal with muliticlass problems [12]. In essence, the approach represents a multi-class classification problem as a set of binary classification problems based on a binary tree structure, which is built recursively by splitting classes into two groups. In Figure 1, we show a situation where a group of five classes is decomposed into three different binary trees, where each tree represents a different set of two-class problems. There are criteria as how to create those binary trees. For instance, if using some criterion we rank the five classes, from the "best" to the "worst", into Class 1, Class 2, Class 3, Class 4, and Class 5, then Figure 1 (a) represents a random decomposition of the classes, while Figure 1 (b) is to separate the best class from the rest at each internal node, and Figure 1 (c) is to select the worst class first each time. These criteria represent a user's view on the priorities of individual classes. It is easy to see that different criteria can create different binary classification trees. One of the possible ways to conduct the overall classification from such a tree is to combine the results of the two-class classifiers using ensemble approaches.

It is often hard to argue convincingly which tree is more advantageous over the other. Then, forming an ensemble using all of them is, among others, a "natural" choice, thus obtaining an ensemble of ensembles. For example, Frank and Kramer use a majority vote to combine randomly generated dichotomy trees [12], where the winning class is the one that is predicted by most of the individual classifiers in the ensemble. Moreover, ensemble approaches aim to combine a set of individual (base) classifiers in such a way as to achieve better classification accuracy than any individual one [3, 13].

3. PROBLEM MOTIVATION AND OUR APPROACH

Frank and Kramer [12] show promising results in various classification tasks using dichotomy-based classification on datasets from the UIC repository. [2] For instance, they show that for the dataset *anneal* with six (6) classes, the accuracy of prediction is as high as 99.33%, while with the dataset *arrhythmia* with 16 classes the reported accuracy is 58.48%, a significant increase over the individual classifier such as *Logistic Regression*. Even with the dataset *letter* with 26 classes, the accuracy is as high as 76.12%. While such highly accurate predictions could be attributed to the small-sized datasets involved, the results are very encouraging, making us wonder whether we could achieve similar accuracy in music genre classification.

3.1 Binary Class Decomposition

In the literature, there are several successful ways to form an ensemble of binary classifiers by decomposing the class space of a multiclass problem into a set of two-class problems. The two common approaches are *One-vs-All* and *One-vs-One*, respectively denoted as *OvA* and *OvO*. In OvA we form an ensemble of n binary classifiers for a given n-class problem. Each classifier is trained to distinguish one class from the rest, and there is one such classifier per class in the ensemble. The OvO approach considers each possible pair of classes in turn and the ensemble consists of $n(n-1)/2$ unique binary classifiers. The classification of a new instance, for example, can be done by simple majority voting where the new instance is predicted as the class whose label was picked by the majority of the binary classifiers. It is not always obvious whether OvA or OvO is the better choice [14]; ergo, we include both in this paper.

Another way to split a set of classes into two is *Group-vs-Group*, denoted as *GvG*. However, such a binary classifier does not say anything definitive about any individual class label in an n-class problem where n is greater than 2. Therefore, to get information about individual classes, we can form a tree of GvG classifiers, denoted as a *Nested Dichotomy Tree* (*NDT*). Such a tree consists of binary classifiers at the internal nodes and individual class labels at the leafs. Starting at the root node, we split the set of all classes into two subsets and train a binary classifier to distinguish between the two. Then we create two children nodes from the subsets in the same manner. This continues until the leaves are created.

This brings us to the *Ensemble of Nested Dichotomies (END)* approach formulated by Frank and Kramer [12]. Here, the ensemble consists of a set of NDTs generated randomly, for example, by randomly choosing the two non-intersecting and jointly exhaustive subsets of classes when splitting an internal node. The final class prediction is selected by majority voting from all of the random NDTs generated in this way.

Because some sets of NDTs yield better classification accuracy than the others, it would be interesting to find a way to pick the best of all sets. However, this task is not trivial

[2] http://archive.ics.uci.edu/ml/

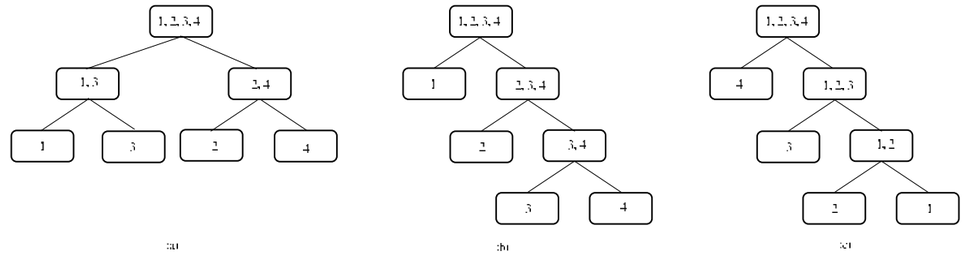


Figure 1. Three different binary trees when decomposing five classes.

and we need to begin by looking at the individual NDTs first. Below, we discuss different NDT structures and ways of forming them heuristically.

3.2 Splitting Criteria in Dichotomies

One splitting criterion of an NDT, as mentioned above, is to randomly choose one or more classes to become Group-1 and the remainder to become Group-2 and train the binary classifier to distinguish between the two groups [12]. Another criterion is to randomly pick exactly half of the classes to become Group-1, which ensures that the resulting NDT is a balanced tree [15].

A non-random criterion would be more interesting. For example, Duarte-Villaseñor *et al.* [16] use clustering to decide the two groups. For our work, we base our criterion on observations about people by noticing that when people are asked to classify an object into a set of categories, two situations arise. First, the more categories to choose from, the longer and "harder" the choice is and mistakes are more frequent. Second, people frequently use the process of elimination, especially when the decision is hard to make. Following this process, the "easy" decisions are made first, and the most difficult ones are left to the last.

Therefore, we propose to rank classes by determining which ones are easier to distinguish from the others and which are harder. To do so, we first use the base classifier with a hold-out subset of training data to solve a multiclass problem and produce a confusion matrix. We believe that when a certain class has a high precision and recall when compared to the other classes, the base classifier recognizes that class better. From this, it follows that the class with the highest score is most distinguishable.

3.3 Proposed Heuristics for Splitting at Each Node

The space of all possible NDTs is too large to be explored exhaustively [12]. We propose four heuristics to construct different NDTs based on our proposed criterion outlined above.

The first one, called *ordered NDT*, separates one class from the rest at each internal node. The choice of which class to separate is based on the order of classes obtained after ranking them using our criterion. It could be either the first one (most distinguishable) or the last one (least distinguishable) and we denote the two *MD-NDT* and *LD-NDT* respectively. This heuristic generates a perfectly imbalanced tree, which is essentially a list of OvR classifiers. MD-NDT represents the observation from people, where we use the process of elimination starting with the class that we are most familiar with. LD-NDT represents the view that if the most confusing class is removed, then it is highly probable that the remaining classes could have a better distinction among them.

The second heuristic generates a *balanced NDT* by following the same intuition as above. We pick half of the classes to split from the rest based on the ranking obtained through our criterion. We then simply split the first half away from the second. In other words we separate half of the classes that are the most distinguishable from the other half that are the least distinguishable. Let us call this heuristic *B1-NDT*.

The third heuristic, *B2-NDT*, also generates a balanced NDT. Here, we are motivated by the intuition that the least distinguishable classes could be easier to distinguish from the most distinguishable ones. So, to fill the first group of classes, we pick alternatively from the classes at the front and the back of the ranked list, and the second group consists of classes from the middle of the list.

During our work with ensembles, we observe that combining many weak classifiers in an ensemble usually results in a stronger classifier. Thus, we combine MD-NDT, LD-NDT, B1-NDT, and B2-NDT using majority vote and denote this ensemble *ENDT* (in contrast to WEKA's END).

4. EXPERIMENT RESULTS AND DISCUSSIONS

In our experiments, we use two benchmark datasets: the *Latin Music Database*, (*LMD*) [17], and the *Million Song Dataset Benchmarking* (*MSDB*) [18]. The LMD dataset, denoted as D_{LMD}, is a carefully constructed dataset that contains equal number of instances per class (300) and each set of class instances is divided equally into training (150) and testing (150) subsets. The MSDB dataset, on the other hand, is much larger. For benchmarking purposes, it includes predetermined designations of instances into training and testing subsets. We use these designations in our experiments with a further requirement that there be at least 988 instances per class designated for training and at least 500 instances for testing. We select at least that many training and testing instances from the respective predesignated pools at random and form D_{MSDB}, which contains 17 genres (classes). For benchmarking purposes, various content-based features are previously extracted and made publicly available for both datasets by their creators.

Due to the way that NDTs are constructed, it is possible that each binary classifier at internal node may have been

trained on imbalanced data, especially in the case of random NDTs. In our experience with confusion matrices, we observed that imbalanced data may skew a given classifier, making it become biased towards the class that is represented with more data. To deal with this situation, we rebalance the data at each internal node every time that the imbalance occurs. This is especially relevant in the case of ordered NDT. For instance consider the case of D_{MSDB}, where at the top node of either MD-NDT or LD-NDT trees, one group is represented by 1000 instances and the other by 16000 instances, approximately. We re-balance via undersampling while also maintaining balanced class representation within each group.

In our experiments, we use WEKA's [3] implementation of four base classifiers outlined below. Additionally we use all of the default parameter settings provided by WEKA. We note that that adjusting each base classifier's parameters would affect the final accuracy, and default settings are likely not the best. However, our task here is to compare different ensemble heuristics on a common ground, leaving the parameters unchanged serves this purpose.

Support Vector Machine, denoted as *SVM*, constructs a hyper-plane or set of hyper-planes in a high-dimensional space and is particularly useful for classification. Given training data with a set of classes, intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest data point of any class. The second classifier in our experiments, *Naive Bayes Classifier (NB)*, is a simple probabilistic classifier based on the Bayesian principle and is particularly suited when dimensionality of the input is high. The third classifier, *k-Nearest Neighbor (k-NN)*, takes training data as vectors in a multidimensional feature space, each with a class label. An unlabeled data point is classified by assigning the label which is most frequent among the k training samples nearest to that point. Fourth, *Logistic Regression (LR)*, measures the relationship between the categorical dependent variable and one or more independent variables. Formal discussions on these classifiers are found in the work of Tsoumakas *et al.* [20].

In addition, we have found through our experiments that between the precision and recall evaluation measures for classification accuracy, the combination of the two works well most of the time for determining the ranking of genres at each NDT node. The few times when a tie occurs, we use precision as the tie-breaker.

4.1 Results and Discussions

We now show the results of our experiments and attempt to analyze them. With the D_{LMD} dataset, we obtain the best classification performance with SVM (WEKA's SMO), followed by k-NN (WEKA's iBK). Hence we only show the SVM results for D_{LMD}. However, D_{MSDB} showed best results with LR followed by SVM. Therefore, for D_{MSDB}, we show the results for LR. In addition to the results of base classifiers and our proposed dichotomy structures, we include the results of WEKA's implementation of OvA, OvO, and the three END ensembles.

Quite unexpectedly, none of the aforementioned ensembles performs significantly better than the base classifiers, as can be seen in Figures 2 and 3, as compared to the results reported in [12] on the UCI datasets. This is against our original intuition and, so far, we do not have a totally convincing explanation for this observation. An initial investigation through probabilistic estimates [12] reveals that dichotomy-based approaches strongly rely on the structures of the binary trees involved. When classifying process reaches the leaves, where the class of a new instance is decided, the probability this happens is conditional on all the probabilistic estimates from the root to the class leaf. We believe that, due to the highly subjective and ambiguous nature of musical genres, the probabilistic estimates are lower and therefore cause lower classification accuracy in music genres at each level of the dichotomy tree. However, we need to explore this through more rigorous probabilistic analysis. This will be our next task in the near future.

[3] http://www.cs.waikato.ac.nz/ml/weka/ [19]

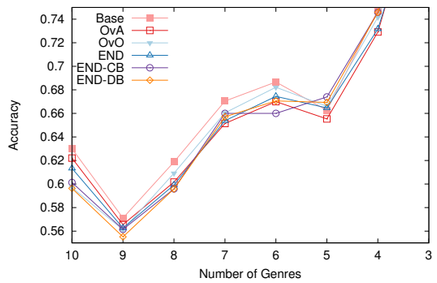


Figure 2. Results of base classifier and WEKA ensembles on D_{LMD}.

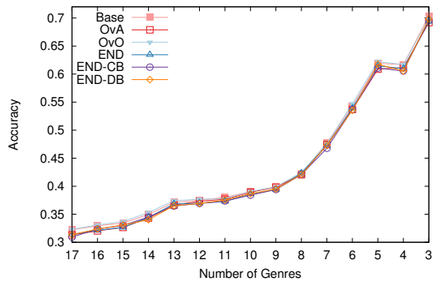


Figure 3. Results of base classifier and WEKA ensembles on D_{MSDB}.

Table 1. A Closer Look at Figure 3 (D_{MSDB})

	Base Classifier	WEKA OvA	WEKA OvO	WEKA END	NDT END-CB	NDT END-DB
17 gen.	0.322	0.313	**0.323**	0.314	0.309	0.314
16 gen.	0.329	0.320	**0.331**	0.321	0.324	0.324

Figures 4 and 5 depict how well our proposed NDTs perform when compared to their base classifiers. Unfortunately, neither one achieves higher accuracy. However, often ENDT performs better than either one of its four constituent classifiers. We conjecture that including additional, good NDTs could improve the ensemble's accuracy beyond that of END, with the advantage of having fewer

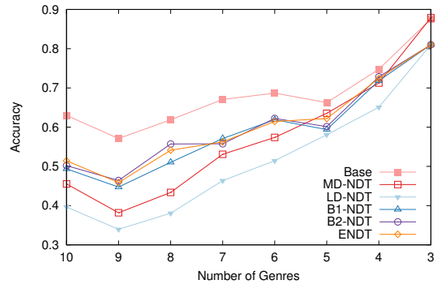


Figure 4. Results of our proposed heuristics on D_{LMD}.

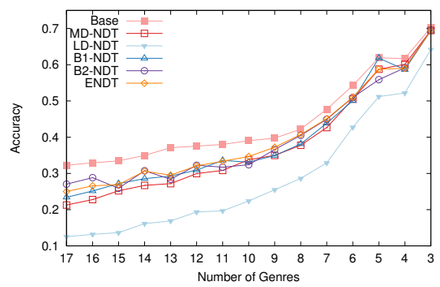


Figure 5. Results of our proposed heuristics on D_{MSDB}.

classifiers in ENDT, making it a better candidate than END in terms of computational complexity.

It can be clearly seen in all of the figures in this paper that our heuristics produce stable results. Moreover, we compared the results of using our approach with all four base classifiers against the the base classifiers themselves. Although we do not include the corresponding figures for the sake of space, we report here that there are no sharp fluctuations in the results, additionally the ENDT ensemble is also as stable as its respective base classifiers.

4.2 Comparison of Different NDTs

Throughout our investigation we observe that imbalanced NDTs span a larger accuracy range than the balanced ones. Moreover, balanced NDTs are at the upper range of all NDTs, while imbalanced NDTs sometimes perform far worse than the balanced ones. We compare the performance of our heuristically obtained NDTs to the performance of any individual NDT in a large population of random balanced NDTs. Figures 6 and 7 compare the trees resulting from our proposed NDT heuristics and a population of 20 random balanced NDTs, from which a typical END would be constructed [12]. It can be seen that MD-NDT performs at the top percentile, while LD-NDT performs worse than any other tree by a large margin. This confirms that our approach to ranking genres (based on how distinguishable they are from others) performs consistently well at picking one of the best (or worst) NDTs in terms of accuracy.

We also observe that our proposed B2-NDT heuristic performs best of all. Moreover, it usually performs better than the majority of random NDTs. This could be due to the datasets we use or something else. It would be worthwhile to explore this further by studying additional heuristics.

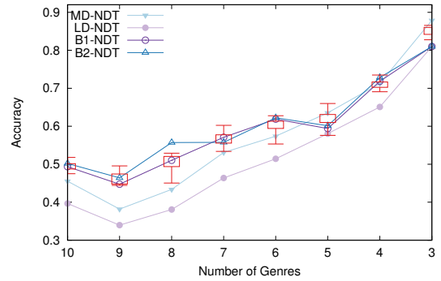


Figure 6. Ranking of our heuristically obtained NDTs as compared to random balanced NDTs using D_{LMD}.

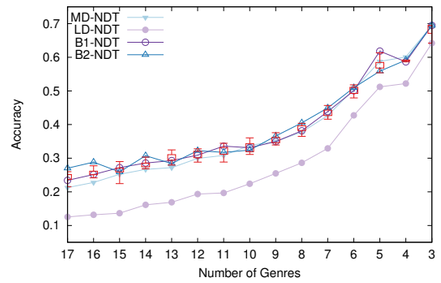


Figure 7. Ranking of our heuristically obtained NDTs as compared to random balanced NDTs using D_{MSDB}.

4.3 Additional Thoughts

We notice throughout our experiments that the rank of a particular genre changes depending on other genres in the set. For example, if we have a ranked order $\{1, 2, 3, 4, 5, 6\}$, after splitting the six classes into two groups and obtaining a new rank for each group, it is possible that the resulting orders could be $\{3, 2, 1\}$ and $\{6, 4, 5\}$. This is why, when we build a dichotomy tree using our approaches, we always find a new ranked order of the genre set at each node before splitting it into two. Whether we can make better use of this observation requires further investigation.

Note that when we gradually increase the number of classes for each dataset in Figure 2, the last genre added to D_{LMD} is *Tango*. This genre happens to be the most differentiable of all genres in the dataset, and therefore, the accuracy of each classifier when classifying 10 genres is actually higher than the accuracy obtained from 9 genres. This can be seen throughout our experiments and in all of the figures presented here. Normally, this would be contrary to the intuition that the classifier accuracy decreases when increasing number of classes. However, our analysis supports this seemingly abnormal behavior and confirms that, in the long run, the increase in the number of classes decreases the classification accuracy.

5. CONCLUSION

In this paper, we have conducted comprehensive empirical experiments to examine various dichotomy-based approaches to genre classification in music. While those approaches result in high classification accuracy in other domains, we show in our experiments that for the majority of variation of the approaches the performance improvement is rather disappointing. Albeit it should be noted that the different heuristic methods discussed are far less than exhaustive. We need more investigative work on dichotomy-based approaches in genre classification in music.

Currently we are considering possible explanations of the unexpected observations in our experiments. We are also conducting more experiments to further verify the results in this paper. For our future work, we will pursue further on probabilistic analysis of dichotomy-based approaches and attempt to explore as why those approaches do not perform well as expected. In addition, we plan to use some other ranking algorithms, such as clustering-based approaches, to rank musical genres when building the dichotomy trees. We will see whether these attempts would help increase genre classification accuracy in music.

6. REFERENCES

[1] T. Li, O. Mitsunori, and G. Tzanetakis, Eds., *Music Data Mining*. CRC Press, 2012.

[2] T. Arjannikov and J. Z. Zhang, "An Empirical Study on Structured Dichotomies in Music Genre Classification," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 493–496.

[3] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, the third ed. Morgan Kaufmann Publishers Inc., 2011.

[4] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, July 2002.

[5] A. Meng and J. Shawe-Taylor, "An Investigation of Feature Models For Music Genre Classification Using the Support Vector Classifier," in *Proceedings of the International Society for Music Information Retrieval Conference*. ISMIR, 2005, pp. 604–609.

[6] M. Li and R. Sleep, "Genre Classification Via an LZ78-based String Kernel," in *Proceedings of the International Society for Music Information Retrieval Conference*. ISMIR, 2005, pp. 252–259.

[7] C. DeCoro, Z. Barutcuoglu, and R. Fiebrink, "Bayesian Aggregation For Hierarchical Genre Classification," in *Proceedings of the International Society for Music Information Retrieval Conference*. ISMIR, 2007, pp. 77–80.

[8] A. Anglade, R. Ramirez, and S. Dixon, "Genre Classification using Harmony Rules Induced from Automatic Chord Transcriptions," in *Proceedings of the International Society for Music Information Retrieval Conference*. ISMIR, 2009, pp. 669–674.

[9] J. C. N. Silla, A. L. Koerich, and C. A. Kaestner, "A Machine Learning Approach to Automatic Music Genre Classification," *Journal of the Brazilian Computer Society*, vol. 14, no. 3, pp. 7–18, 2008.

[10] C. Sanden and J. Z. Zhang, "An Empirical Study of Multi-label Classifiers for Music Tag Annotation," in *Proceedings of the International Society for Music Information Retrieval Conference*. ISMIR, 2011, pp. 717–722.

[11] B. L. Sturm, "A Survey of Evaluation in Music Genre Recognition," in *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, vol. 8382, pp. 29–66.

[12] E. Frank and S. Kramer, "Ensembles of Nested Dichotomies for Multi-class Problems," in *Proceedings of the Twenty-first International Conference on Machine Learning*. ACM, 2004, pp. 39–46.

[13] N. V. Chawla and S. J., "Exploiting diversity in ensembles: improving the performance on unbalanced datasets," in *Proceedings of International Conferernce on Multiple Classifier Systems*, 2007, pp. 397–406.

[14] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *The Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[15] L. Dong, E. Frank, and S. Kramer, "Ensembles of balanced nested dichotomies for multi-class problems." in *Knowledge Discovery in Databases: PKDD 2005*. Springer Berlin Heidelberg, 2005, pp. 84–95.

[16] M. M. Duarte-Villaseñor, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and M. Flores-Garrido, "Nested Dichotomies Based on Clustering," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer Berlin Heidelberg, 2012, vol. 7441, pp. 162–169.

[17] C. N. J. Silla, A. L. Koerich, and C. A. Kaestner, "The Latin Music Database," in *Proceedings of the International Society for Music Information Retrieval Conference*. ISMIR, 2008, pp. 451–456.

[18] A. Schindler, R. Mayer, and A. Rauber, "Facilitating Comprehensive Benchmarking Experiments on the Million Song Dataset," in *Proceedings of the International Society for Music Information Retrieval Conference*. ISMIR, 2012, pp. 469–474.

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Exploration Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[20] G. Tsoumakas, I. Katakis, and I. Vlahavas, *Mining multi-label data. Data Mining and Knowledge Discovery Handbook*. Springer Link, 2010.

Algorithmic Composition Parameter as Intercultural and Cross-level MIR Feature: The Susceptibility of Melodic Pitch Contour

Hsin-Ming Lin
Department of Music
University of California, San Diego
hsl040@ucsd.edu

Shlomo Dubnov
Department of Music in affiliation with
Department of Computer Science Engineering
University of California, San Diego
sdubnov@ucsd.edu

ABSTRACT

Algorithmic composition (AC) and music information retrieval (MIR) can benefit each other. By compositional algorithms, scientists generate vast materials for MIR experiment; through MIR tools, composers instantly analyze abundant pieces to comprehend gross aspects. Although there are manifold musicologically valid MIR features, most of them are merely applicable to Western music. Besides, most high-level and low-level features are not interchangeable to retrieve from both symbolic and audio samples. We investigate the susceptibility of melodic pitch contour, a parameter from an AC model. It was created to regulate a generative monophonic melody's sensitivity to make a return after consecutive pitch intervals. It takes audio frequency values rather than symbolic pitch numbers into consideration. Hence we expect its intercultural and cross-level capabilities. To validate, we modify the original model from compositional to analytical functions. Our experimental results unveil a clear trend of mean susceptibilities from vocal to instrumental styles in 16522 samples from 81 datasets across numerous composers, genres, eras, and regions. We demonstrate the mutual benefits between AC and MIR. The parameter operates as an intercultural and cross-level feature. The relationship between susceptibility and register width is surprising in several comparisons. Further investigation is ongoing to answer more questions.

1. INTRODUCTION

1.1 Mutualism

"Music is an ever-evolving field of artistic and scientific expression." [1] Composers have been devising algorithms or computer programs to manipulate musical ingredients. [2, 3] Meanwhile, engineers have been implementing music information retrieval (MIR) tools to retrieve various features from music [4]. Thanks to such advance in technology, nowadays composers may avail themselves of the opportunity to analyze enormous amount of pieces.

On the other hand, MIR can benefit from algorithmic composition (AC). AC is able to generate limitless free pieces with explicit ground truth [5]. That can promote even better MIR techniques, which again expand composer's point of view to design creative compositional algorithms.

1.2 Versatility

Despite of the aforementioned mutualism, prevalent MIR applications and computational analyses greatly rely on Western music theories (e.g. [6]). Many high-level features (e.g. [7]) are incompatible with non-Western music. Additionally, promising intercultural statistical universals (e.g. [8]) are almost too general to adopt in MIR. Therefore, current musical features applicable to multiple cultures (not necessarily universal) are remarkably inadequate.

Moreover, high-level features are "hard to extract" [4] especially from audio data, while low-level features do not exist in symbolic data. Features across "symbolic" and "sub-symbolic" levels [9] are lacking. Combining features extracted from separate sources (e.g. [10]) is, however, not always practical. With intent to seek for intercultural and cross-level features, we investigate an AC model below.

1.3 Singability

In a previous AC program, the author conceived a particular parameter and coined its name "susceptibility", a nod to the magnetic susceptibility in electromagnetism. The parameter regulated a generative monophonic melody's sensitivity to make a return after successive pitch intervals. [11, 12] It might innately be capable of not only melodic pitch contour control but also the "effect of tessitura" [13] and the "low-skip bias" [14] even without any prior constraint on pitch range. The compositional algorithm is independent of any tuning system because it takes audio frequencies rather than symbolic pitches into consideration. For symbolic notes, users should convert pitch numbers into equivalent frequency values. Thus we conjecture that it has intercultural and cross-level potentialities.

"A melody could hardly include unmelodious elements; […] The nature and technique of the primordial musical

instrument, the voice, determines what is singable. The concept of the melodious in instrumental melody has developed as a free adaptation from the vocal model." [15] Singability of a monophonic melody depends upon but not limited to elements as follows:

1. pitch range
2. prevalence of consonant and dissonant intervals
3. the smallest and the largest interval between pitches
4. average melodic interval size [7]
5. most common melodic interval size [7]
6. comfortable and uncomfortable register
7. global and local sensitivity (susceptibility) to transfer of register

We suppose that the susceptibility is, on average, higher in vocal styles than instrumental ones. In subsequent paragraphs, we revise the original compositional algorithms, implement a program, and collect datasets to verify our hypothesis.

2. METHODS

2.1 Formulas

The original model [12] was invented for AC. Several parameters were adjusted by the user. For this reason, we have to modify it on purpose to retrieve the susceptibility value from a sample. First, the central frequency is now automatically calculated from the pitch range of the pre-existing melody. The audio frequency or its equivalent value of a symbolic pitch according to appropriate tuning system is f. The highest frequency in the melody is f_{max}; the lowest frequency in the melody is f_{min}. The total note numbers of the monophonic melody is n. The register width of the melody is defined as

$$w = \log_2 \frac{f_{max}}{f_{min}} \quad (1)$$

The central frequency of the melody is

$$f_0 = f_{min} \times \sqrt{\frac{f_{max}}{f_{min}}} \quad (2)$$

Second, the part of energy ratio interval is unchanged. The frequency ratio of each successive interval is

$$r_n = \frac{f_n}{f_0};\ n \geq 0 \quad (3)$$

Its energy ratio is

$$e_n = (r_n)^2;\ n \geq 0 \quad (4)$$

The energy ratio interval is

$$i_n = e_n - e_{(n-1)};\ n \geq 1 \quad (5)$$

Third, we need a constant k to enlarge the initial tolerable energy ratio maximum and minimum. We assign the same tentative value (k = 4) to all experiments in this research. The initial maximum and minimum are no longer set in advance by the user. On the contrary, the initial tolerable energy ratio maximum is given by

$$e\max_1 = \max\{e_j\} \times k;\ j = 1,2,3,\ldots,n \quad (6)$$

while the reciprocal of initial tolerable energy ratio minimum can be written as

$$\frac{1}{e\min_1} = \frac{1}{\min\{e_j\} \times k};\ j = 1,2,3,\ldots,n \quad (7)$$

Next, s (susceptibility) is the largest possible value which is approached by means of our retrieval program. The following part is similar to the original. The tolerable energy ratio maximum is defined as

$$e\max_n = e\max_{n-1} - (i_{n-1} \times s);\ n \geq 2 \quad (8)$$

while the reciprocal of tolerable energy ratio minimum is

$$\frac{1}{e\min_n} = \frac{1}{e\min_{n-1}} + (i_{n-1} \times s);\ n \geq 2 \quad (9)$$

The tolerable frequency ratio maximum is

$$r\max_n = \sqrt{e\max_n};\ n \geq 1 \quad (10)$$

The reciprocal of tolerable frequency ratio minimum is

$$\frac{1}{r\min_n} = \sqrt{\frac{1}{e\min_n}};\ n \geq 1 \quad (11)$$

i.e.

$$r\min_n = \sqrt{e\min_n};\ n \geq 1 \quad (12)$$

Finally, we revise the last part for better visualization. The logarithmic frequency ratio is defined as

$$l_n = \log_2(r_n);\ n \geq 0 \quad (13)$$

The logarithmic tolerable frequency ratio maximum is

$$l\max_n = \log_2(r\max_n);\ n \geq 1 \quad (14)$$

The logarithmic tolerable frequency ratio minimum is

$$l\min_n = \log_2(r\min_n);\ n \geq 1 \quad (15)$$

2.2 Example

In order to exemplify, we select a short melody from counterpoint textbook [16] to process (see table 1 and figure 1). The melody is notated in symbolic pitch, so we have to assign audio frequency to every note. For the purpose of simple values and straightforward calculation, we avoid adopting the popular twelve-tone equal temperament. Nevertheless, users are allowed to refer to any proper tuning system for conversion from pitches into frequencies.

In this example, the frequency of the first pitch (f_1) is 240 Hz. The highest frequency (f_{max}) is 360, while the lowest frequency (f_{min}) is 240 Hz. As a result, the central frequency (f_0) is about 294 Hz. The susceptibility is approached from zero to the largest possible value using an increment of 0.01. When the susceptibility reaches 10.32, the e_3 will exceed $e\max_3$. In consequence, the final retrieved susceptibility is 10.31.

2.3 Program

We implement a retrieval program in Python with the music21 [17] version 2.2.1. The music21 toolkit is able to parse several symbolic file formats and to represent a "score > part > measure > voice > chord or note" hierarchy. It has the capability to correctly extract specific part and voice whenever the information is accessible in the sample file. If there is any chord in individual voice, we simply extract the highest note of the chord. For any polyphonic sample, we extract highest notes of the first voice in its top part unless otherwise specified.

Some datasets barely include audio frequency annotations instead of symbolic files. In this situation, our program directly reads the frequency values without conversions by means of the music21 toolkit.

Pitch	La 3	Mi 4	Si 3	Re 4	Do 4	Si 3	La 3
f (f_0≈294)	240	360	270	320	288	270	240
r (r_0=1)	0.816	1.225	0.919	1.089	0.980	0.919	0.816
e (e_0=1)	0.667	1.500	0.844	1.185	0.960	0.844	0.667
i	-0.333	0.833	-0.656	0.341	-0.225	-0.116	-0.177
e-max	6.000	9.437	0.845	7.611	4.091	6.412	7.611
1/e-min	6.000	2.563	11.155	4.389	7.909	5.588	4.389
e-min	0.167	0.390	0.090	0.228	0.126	0.179	0.228
r-max	2.449	3.072	0.919	2.759	2.023	2.532	2.759
1/r-min	2.449	1.601	3.340	2.095	2.812	2.364	2.095
r-min	0.408	0.625	0.299	0.477	0.356	0.423	0.477
l (l_0=0)	-0.292	0.292	-0.123	0.123	-0.029	-0.123	-0.292
l-max	1.292	1.619	-0.121	1.464	1.016	1.340	1.464
l-min	-1.292	-0.679	-1.740	-1.067	-1.492	-1.241	-1.067

Table 1. Retrieval Example. susceptibility = 10.31.

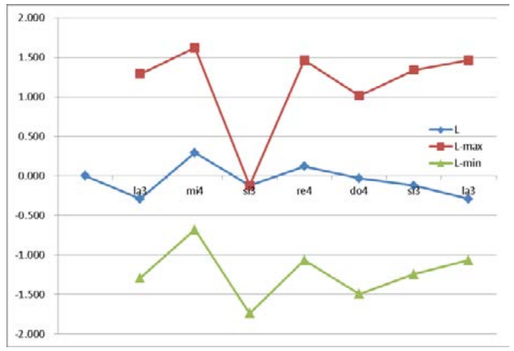


Figure 1. Melodic Pitch Contour (middle line) in the Retrieval Example. susceptibility = 10.31.

3. DATASETS

3.1 Sources

We collect datasets from four sources. Most samples come from KernScores [18]. Its genres range from monophonic and harmonized songs to classical instrumental music. Vocal samples cover folk melodies from four continents (Europe, Asia, North America, and South America), Bach chorales, and early Renaissance pieces compiled in Josquin Research Project [19]; instrumental samples comprise string quartets, piano sonatas, Mazurkas, and preludes.

The second largest dataset is SymbTr [20], a Turkish Makam music symbolic representation database, which consists of instrumental and vocal pieces. The third one is The Aria Database [21]. The website preserves rich information about opera and operatic arias. We download all available 197 aria MIDI files. The last dataset, MedleyDB [22], incorporates 9 genres (rock, pop, classical, jazz, rap, fusion, world/folk, musical/theater, and singer/songwriter). It does not have symbolic music files but three types of melody annotation based on different definitions. We choose the first one in which the fundamental frequency curve of the predominant melodic line is drew from a single source.

3.2 Cleaning

In our initial retrieval, we notice that several datasets and samples have illogical values such as extremely large register width. There are also problematic files which are incomplete, separate one-part notes into two parts, or mix multiple parts' notes in one part or MIDI track. We scrutinize every suspicious file and discard all unacceptable ones. Furthermore, we omit some symbolic files which the music21 toolkit cannot parse. In the end, we retrieved features from 16522 samples (see table 2 and appendix).

Source	Instrumental Dataset(s)	Vocal Dataset(s)	Instrumental Samples	Vocal Samples
Aria[21]	0	1	0	177
KernScores[18]	21	55	1857	12360
MedleyDB[22]	1	1	47	61
SymbTr[20]	1	1	187	1833
total	23	58	2091	14431

Table 2. Quantities of Retrieved Samples.

4. RESULTS

4.1 Trends

Our experiments reveal a clear trend of mean susceptibilities from vocal to instrumental styles (see figure 2). All 57 datasets with larger mean susceptibilities than "Aria" are vocal except "Keyboard\Mazurka"; all 23 datasets with smaller mean susceptibilities are instrumental except "MedleyDB\Vocal" (see appendix). The ranges of susceptibilities in vocal datasets are broadly higher and wider, while they are mostly lower and narrower in instrumental datasets.

Results obviously illustrate the effect of tessitura [13]. It forces melodies which have smaller register widths to have larger susceptibilities. By contrast, melodies which have larger register widths have more flexibility. Most retrieved samples congregate along a nonlinear trend line, while some scatter in upper areas (see figure 3). The distribution overlap between vocal and instrumental samples is quite reasonable since composers may more or less deploy vocal composition strategies in both vocal and instrumental pieces. After all, people prefer to keep somewhat singability even in instrumental melodies.

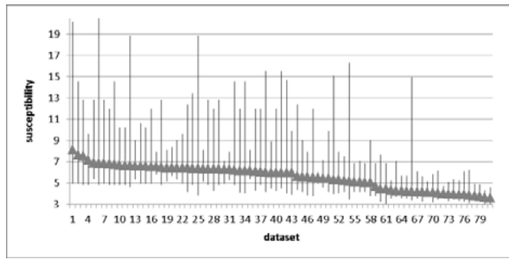


Figure 2. Susceptibility Ranges and Means of Each Dataset. The 58th dataset is "Aria".

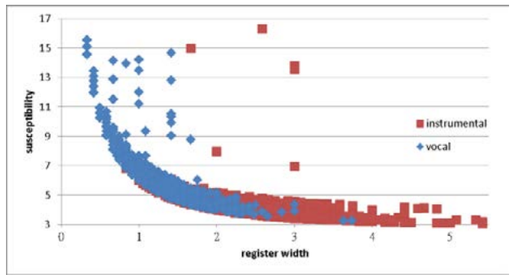


Figure 3. Distribution of Susceptibilities and Register Widths of the 16518 of 16522 Samples. correlation coefficient ≈ -0.77 (both); -0.54 (instrumental); -0.85 (vocal).

4.2 Cases

After further inspection, we find more noticeable relationships between susceptibility and register width. For example, the "SymbTr\Instrumental" dataset has a considerable negative correlation between susceptibility and register width (see figure 4), while the "Keyboard\Mazurka" dataset has virtually no correlation (see figure 5). Nonetheless, they are two dissimilar genres. We have to compare datasets in the same genre, too.

The correlations between susceptibility and register width from three classical composers are distinct across each part in all their string quartets (see table 3). In addition, the exact sequence also exists in their piano sonatas, (see table 4). As we anticipated, the correlation is more significantly negative. String instrument players often have less difficulty to perform large interval skips than keyboard players. One can tell this characteristic through those distinctive correlation coefficients.

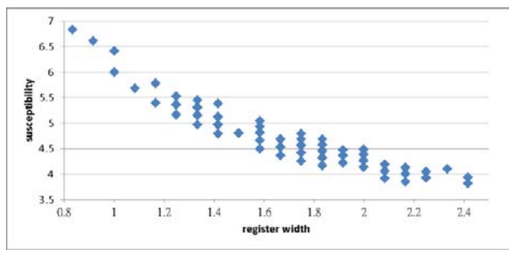


Figure 4. Distribution of Susceptibilities and Register Widths of the samples in "SymbTr\Instrumental" dataset. correlation coefficient ≈ -0.94; p-value < 0.001.

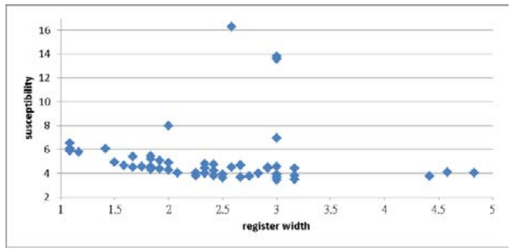


Figure 5. Distribution of Susceptibilities and Register Widths of the samples in "Keyboard\Mazurka" dataset. correlation coefficient ≈ -0.05; p-value ≈ 0.713.

Dataset	Correlation Coefficient	P-value
Mozart\Vn-2	-0.746893759	< 0.001
Mozart\Vc	-0.735371459	< 0.001
Mozart\Va	-0.726169124	< 0.001
Mozart\Vn-1	-0.693441731	< 0.001
Haydn\Va	-0.674282865	< 0.001
Haydn\Vc	-0.631095435	< 0.001
Haydn\Vn-2	-0.595389532	< 0.001
Haydn\Vn-1	-0.48000777	< 0.001
Beethoven\Vc	-0.450330236	< 0.001
Beethoven\Vn-2	-0.422600441	≈ 0.002
Beethoven\Vn-1	-0.359701009	< 0.001
Beethoven\Va	-0.324189128	≈ 0.006

Table 3. Correlations Between Susceptibilities and Register Widths in Each Part of String Quartets

Dataset	Correlation Coefficient	P-value
Mozart Sonata	-0.899077571	< 0.001
Haydn Sonata	-0.819956948	< 0.001
Beethoven Sonata	-0.612629034	< 0.001

Table 4. Correlations Between Susceptibilities and Register Width in Each Part of Piano Sonatas

5. DISCUSSION

5.1 Application

Given that the intercultural correlations between susceptibility and register width in instrumental or vocal samples respectively are common principles, one can consider the deviation distance to be a degree of novelty. The nearly 30 "outliers" in the distribution come from 13 sub-datasets in at least 5 genres. 6 of them are instrumental samples. Thereby, the white area above the cluster is viable but barely exploited by human (see figure 3). Composers may attempt handling related parameters to generate unusual materials.

AC is usually being applied in two ways: "generating music imitating a corpus of compositions or a specific style" and "automating composition tasks to varying degrees." [3] The other contribution is the byproduct like innovative parameters which are engaged as MIR features. The susceptibility is retrievable from either symbolic or sub-symbolic monophonic pitches in any tuning system. Such intercultural and cross-level features are indispensable for dealing with diverse music from all over the world.

5.2 Expectation

In the original AC model, the central frequency is designated by the user. The register width is inessential in terms of the regulation of generative melodic pitch contour. Nevertheless, the susceptibility is certainly not a sole feature to discriminate vocal and instrumental samples. Still, it is a unique perspective to appreciate styles. We hope to further explain the fascinating distinction of correlation between susceptibility and register width. The finding tells us that the correlation between features could serve as a better feature in some circumstances.

Traditionally, features are the global characters of a sample (e.g. [7]). Although so-called local high-level features [23] and string methods [24] have been proposed, the internal relationships between different features inside a sample are insufficiently examined. If people treat a dataset like a big sample, then each piece is a section of the sample. If they want to retrieve the correlations between features within a piece, they must divide the piece into segments. The susceptibility is, however, a cumulative factor due to its original design. Local or sectional susceptibility may not be a musicologically valid feature. We have another ongoing investigation to answer more questions.

5.3 Conclusion

We modify the previous composition model to analyze samples across numerous composers, genres, eras, and regions. We confirm that the mean susceptibility is higher in vocal datasets than instrumental. The ranges of vocal susceptibilities are broadly higher and wider than instrumental.

The correlations between susceptibilities and register widths in each dataset are surprising in some comparisons. We need to find solutions for meaningful local or sectional features. Above all, we demonstrate the mutual benefits between composers and scientists. The AC parameter, susceptibility, operates as an intercultural and cross-level MIR feature. The experimental results provide feedback to composers. They may create new parameters and algorithms to search unknown dimensions and "space of that which might be explored." [25]

Acknowledgments

The first author would like to thank Dr. Yi-Hsuan Yang and Dr. Li Su in Music and Audio Computing Lab, Research Center for Information Technology Innovation, Academia Sinica (Taipei, Taiwan) for hiring him, enlightening him on MIR, and support for his related research in past summers and years.

6. REFERENCES

[1] G. Mazzola, J. Park, and F. Thalmann. *Musical Creativity: Strategies and Tools in Composition and Improvisation*. Springer, 2011.

[2] G. Nierhaus. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer, 2009.

[3] J. D. Fernández and F. Vico, "AI Methods in Algorithmic Composition: A Comprehensive Survey," *Journal of Artificial Intelligence Research*, vol. 48, pp. 513–582, 2013.

[4] M. A. Casey et al., "Content-based Music Information Retrieval: Current Directions and Future Challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[5] B. L. Sturm and N. Collins, "The Kiki-Bouba Challenge: Algorithmic Composition for Content-based MIR Research & Development," *Proceedings of the International Symposium on Music Information Retrieval*, Taipei, 2014, pp. 21–26.

[6] D. Meredith. *Computational Music Analysis*. Springer, 2016.

[7] C. McKay, *Automatic Genre Classification of MIDI Recordings*, Dissertation, McGill University, 2004.

[8] P. E. Savage et al., "Statistical Universals Reveal the Structures and Functions of Human Music," *Proceedings of the National Academy of Sciences*, vol. 112, no. 29, pp. 8987–8992, 2015.

[9] R. Rowe, "Split Levels: Symbolic to Sub-Symbolic Interactive Music Systems." *Contemporary Music Review*, vol. 28, no. 1, pp. 31–42, 2009.

[10] C. McKay and I. Fujinaga. "Combining Features Extracted from Audio, Symbolic and Cultural Sources," *Proceedings of the International Symposium on Music Information Retrieval*, Philadelphia, 2008, pp. 597–602.

[11] H.-M. Lin and C.-F. Huang. "An Algorithmic Composition Method of Monophonic Pitches Based on the Frequency Relationship of Melody Motion," *Proceedings of the International Computer Music Conference*, New York, 2010.

[12] H.-M. Lin. *Algorithmic Composition of Monophonic Pitches Based on Conservation of Energy in Melodic Contour*. Master's thesis, National Chiao Tung University, Hsinchu, Taiwan, 2011.

[13] P. von Hippel and D. Huron. "Why Do Skips Precede Reversals: The Effect of Tessitura on Melodic Structure," *Music Perception: An Interdisciplinary Journal*, vol. 18, no. 1, pp. 59–85, 2000.

[14] P. Ammirante and F. A. Russo. "Low-Skip Bias: The Distribution of Skips Across the Pitch Ranges of Vocal and Instrumental Melodies is Vocally Constrained," *Music Perception: An Interdisciplinary Journal*, vol. 32, no. 4, pp. 355–363, 2015.

[15] A. Schoenberg, *Fundamentals of Musical Composition*. Translated by G. Strang. London: Faber & Faber, 1967.

[16] K. Jeppesen, *Counterpoint: The Polyphonic Vocal Style of the Sixteenth Century*. Translated by G. Haydon. New York: Dover Publications, 1992.

[17] M. C. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," *Proceedings of the International Symposium on Music Information Retrieval*, Utrecht, 2010, pp. 637–642.

[18] C. S. Sapp, "Online Database of Scores in the Humdrum File Format," *Proceedings of the International Symposium on Music Information Retrieval*, London, 2005, pp. 664–665.

[19] The Josquin Research Project, http://josquin.stanford.edu

[20] M. K. Karaosmanoğlu, "A Turkish Makam Music Symbolic Database for Music Information Retrieval: SymbTr," *Proceedings of the International Symposium on Music Information Retrieval*, Porto, 2012, pp. 223–228.

[21] The Aria Database, http://www.aria-database.com

[22] R. M. Bittner et al., "MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research," *Proceedings of the International Symposium on Music Information Retrieval*, Taipei, 2014, pp. 155–160.

[23] P. van Kranenburg, A. Volk, and F. Wiering, "A Comparison between Global and Local Features for Computational Classification of Folk Song Melodies," *Journal of New Music Research*, vol. 42, no. 1, pp. 1–18, 2013.

[24] R. Hillewaere, B. Manderick, and D. Conklin. "String Methods for Folk Tune Genre Classification," *Proceedings of the International Symposium on Music Information Retrieval*, Porto, 2012, pp. 217–222.

[25] R. Reynolds. "The Evolution of Sensibility," *Nature*, vol. 434, pp. 316–319, 2005.

7. APPENDIX

7.1 Mean Susceptibilities in All 81 Datasets

Dataset	Samples	Susceptibility	STD
Mono\EFC\Europa\jugoslav	115	8.13	2.50
Mono\EFC\Europa\deutschl\kinder	213	7.62	2.08
Mono\friuli (Italy)	80	7.54	1.88
Mono\EFC\Europa\deutschl\variant	26	7.17	1.11
Mono\EFC\Europa\czech	43	6.88	1.51
Mono\EFC\Europa\schweiz	93	6.86	2.82
Mono\EFC\Europa\rossiya	37	6.85	1.66
Mono\EFC\Europa\nederlan	85	6.81	1.24
Mono\EFC\Europa\deutschl\altdeu1	309	6.72	1.28
Bach\185 Chorales\Alto	185	6.67	1.17
Bach\371 Chorales\Alto	370	6.63	1.05
Mono\EFC\Europa\deutschl\altdeu2	316	6.62	1.20
Mono\EFC\Europa\polska	25	6.61	1.10
Bach\371 Chorales\Soprano	370	6.59	1.07
Bach\185 Chorales\Soprano	185	6.59	0.99
Mono\EFC\Europa\magyar	45	6.52	1.52
Mono\EFC\Europa\italia	8	6.52	0.71
Mono\EFC\Europa\elsass	91	6.47	1.51
Mono\EFC\Europa\danmark	9	6.44	0.84
Mono\EFC\Europa\france	14	6.42	0.86
Mono\EFC\Europa\lothring	71	6.42	0.78
Mono\EFC\Europa\romania	28	6.41	1.07
Mono\EFC\Europa\oesterrh	104	6.40	1.55
Mono\EFC\Europa\deutschl\ballad	687	6.40	1.09
Mono\EFC\Europa\deutschl\erk	1700	6.39	1.25
Mono\EFC\Europa\sverige	11	6.34	0.80
Bach\185 Chorales\Tenor	185	6.33	0.92
Mono\EFC\Europa\deutschl\zuccal	616	6.32	1.04
Bach\371 Chorales\Tenor	370	6.30	0.86
Mono\EFC\Europa\luxembrg	8	6.26	0.81
Mono\EFC\Europa\ukraina	13	6.25	0.88
Mono\pentatonic	67	6.21	1.30
Mono\EFC\Europa\misc	30	6.18	1.77
Mono\American\Pawnee	86	6.18	1.96
Mono\EFC\Europa\england	4	6.14	1.14
Mono\EFC\Europa\deutschl\dva	106	6.12	1.24
Mono\EFC\Europa\deutschl\allerkbd	110	6.05	0.96
Mono\EFC\Europa\deutschl\boehme	704	5.99	0.94
Mono\British children's songs	38	5.97	1.05
Mono\EFC\Europa\deutschl\fink	566	5.96	0.91
Mono\Nova Scotia (Canada)	152	5.96	1.24
Mono\EFC\China\han	1222	5.95	1.20
Harmo\Deutscher Liederschatz	231	5.93	0.88
JRP\Secular Song	157	5.62	0.93
Mono\EFC\China\natmin	206	5.58	0.86
Mono\EFC\China\xinhua	10	5.58	1.57
Mono\American\Sioux	245	5.54	1.06
Mono\EFC\Europa\deutschl\test	12	5.46	0.41
Mono\EFC\Europa\tirol	14	5.45	0.76
Mono\EFC\China\shanxi	802	5.42	0.60
SymbTr\Vocal	1833	5.31	0.89
Mono\American\Ojibway	42	5.30	0.88
JRP\Motet	178	5.22	0.59
Keyboard\Mazurka	55	5.14	2.44
Bach\371 Chorales\Bass	370	5.11	0.54
JRP\Mass	411	5.09	0.50
Bach\185 Chorales\Bass	185	5.08	0.52
Aria	177	5.04	0.63
SymbTr\Instrumental	187	4.67	0.52
MedleyDB\Vocal	61	4.43	0.72
Keyboard\Chopin Prelude	24	4.42	0.85
Keyboard\Bach WTC Fugue	44	4.34	0.37
String Quartet\Haydn\Va	201	4.29	0.44
String Quartet\Mozart\Va	82	4.24	0.37
String Quartet\Mozart\Vn-2	82	4.23	0.44
String Quartet\Beethoven\Vn-2	71	4.22	1.34
String Quartet\Haydn\Vn-2	201	4.17	0.38
String Quartet\Mozart\Vc	82	4.15	0.44
String Quartet\Beethoven\Va	71	4.14	0.33
MedleyDB\Instrumental	47	4.12	0.63
String Quartet\Haydn\Vc	201	4.04	0.39
Keyboard\Clementi Sonatina	17	4.00	0.35
Keyboard\Scott Joplin	47	3.97	0.41
String Quartet\Mozart\Vn-1	82	3.94	0.42
Keyboard\Scarlatti Sonata	59	3.91	0.38
String Quartet\Haydn\Vn-1	201	3.89	0.38
Keyboard\Mozart Sonata	69	3.85	0.65
String Quartet\Beethoven\Vc	71	3.81	0.34
String Quartet\Beethoven\Vn-1	71	3.74	0.39
Keyboard\Haydn Sonata	23	3.62	0.39
Keyboard\Beethoven Sonata	103	3.57	0.36

The GiantSteps Project: A Second-Year Intermediate Report

Peter Knees,[1] **Kristina Andersen,**[2] **Sergi Jordà,**[3] **Michael Hlatky,**[4] **Andrés Bucci,**[5] **Wulf Gaebele,**[6] **Roman Kaurson**[7]
[1] Dept. of Computational Perception, Johannes Kepler University Linz, Austria
[2] Studio for Electro-Instrumental Music (STEIM), Amsterdam, Netherlands
[3] Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[4] Native Instruments GmbH, Berlin, Germany [5] Reactable Systems S.L., Barcelona, Spain
[6] Yadastar GmbH, Cologne, Germany [7] JCP-Connect SAS, Rennes, France
info@giantsteps-project.eu

ABSTRACT

We report on the progress of GiantSteps, an EU-funded project involving institutions from academia, practitioners, and industrial partners with the goal of developing new concepts for intelligent and collaborative interfaces for music production and performance. At the core of the project is an iterative, user-centric research approach to music information retrieval (MIR) and human computer interaction (HCI) that is designed to allow us to accomplish three main targets, namely (1) the development of intelligent musical expert agents to support and inspire music makers, (2) more intuitive and collaborative interfaces, and (3) low-complexity methods addressing low-cost devices to enable affordable and accessible production tools and apps. In this paper, we report on the main findings and achievements of the project's first two years.

1. INTRODUCTION

The stated goal of the GiantSteps project is to create the so-called "seven-league boots" for future music production. [1] Built upon an iterative and user-centric research approach to music information retrieval (MIR) and human computer interaction (HCI), the project is developing digital musical tools and music analysis components that provide more intuitive and meaningful interfaces to musical data and knowledge in order to empower music practitioners to use their creative potential. [2] In particular, we want to achieve this by targeting three directions:

1. Developing **musical expert agents**, i.e., supportive systems for melody, harmony, rhythm, or style to guide users when they lack inspiration or technical or musical knowledge.

2. Developing **improved user interfaces** and paradigms for (collaborative) musical human-computer interaction that are easily graspable by novices and lead to unbroken workflows for professionals.

3. Developing low-complexity algorithms for music analysis addressing **low-cost devices** to enable affordable and accessible production tools and apps.

In order to meet these goals, GiantSteps is set up as a transdisciplinary research project, carried out by a strong and balanced consortium including leading music information research institutions *(UPF-MTG, CP-JKU)*, leading industry partners *(Native Instruments, Reactable, JCP-Connect)* and leading music practitioners *(STEIM, Red Bull Music Academy/Yadastar).* [3]

With this consortium, the project aims at combining techniques and technologies in new and unprecedented ways, all driven by users' practical needs. This includes the combination of state-of-the-art interfaces and interface design techniques with advanced methods in music information retrieval (MIR) research that have not yet been applied in a real-time interaction context or with creativity objectives. In addition to this, the industrial partners ensure alignment of the developments with existing market requirements, allowing for a smooth integration of outcomes into real-world systems.

In this report, we describe the findings and achievements of the project within the first two years. Section 2 outlines the user-centric design approach that generates the requirements for the technical developments. Section 3 deals with the advances in MIR that are necessary in order to enable music expert agents, novel visualizations, and new interfaces as discussed in section 4, as well as more effective algorithms for low-resource devices. Section 5 describes the outcomes of the project in terms of market-released products. Section 6 concludes with an outlook to the project's final year and beyond.

2. USER-CENTRIC DESIGN APPROACH

The project's constellation as a collaboration between music research institutions, manufacturers of software and hardware for music production, R&D companies and music practitioners allows us to engage in an ongoing conversation with the professional makers of Electronic Dance

[1] *http://www.giantsteps-project.eu*
[2] Note that parts of this paper have already been published in [1].

[3] *http://redbullmusicacademy.com*; From the webpage: "The Red Bull Music Academy is a world-travelling series of music workshops and festivals *[in which]* selected participants – producers, vocalists, DJs, instrumentalists and all-round musical mavericks from around the world – come together in a different city each year. For two weeks, each group will hear lectures by musical luminaries, work together on tracks and perform in the city's best clubs and music halls."

Figure 1. Impressions from the user sessions: participatory workshop, mock-up prototype for music event live manipulation, visual sound query interface, and practitioners at the Red Bull Music Academy in Tokyo (from left to right).

Music (EDM), that we consider our key users, and to tailor musical tools to their needs. This takes the form of generating user requirements and testing prototypes with end-users in an iterative process throughout the project.

The overall goals of this user involvement are to establish a range of current creative practices for musical expression in EDM, explore mental models of musical qualities, produce user-generated ideas through explorative making, and inspire design and non-design related tasks within the project, cf. [2, 3]. To this end, we conduct a series of different workshop and interview sessions involving expert users (cf. fig. 1). The user sessions comprise interface-specific and work-practice-related interviews and cognitive walkthroughs, e.g., to identify breaks in workflows, as well as ad-hoc, open-ended interviews, carried out on location at STEIM, Native Instruments, Music Hack Days, and the Red Bull Music Academy, resulting in interactions with over 200 individual users so far.

To ensure traceability of the identified goals and requirements throughout the process of developing prototypes, we have set up a system for managing prototypes, generating keyword requirements, and exploring ideas in functional and non-functional prototypes. Fig. 2 illustrates the overall flow of user involvement in the project. From user conversations, we extract the most pertinent ideas as keywords that are either addressed in a concrete technical implementation or — if not yet at that point — a conceptual prototype. Either can be exposed to the user or, in case of a technical prototype, quantitatively evaluated (particularly low-level MIR algorithms). To close the circle, results are evaluated with users, leading to a new round of user conversations informing the project's next iteration.

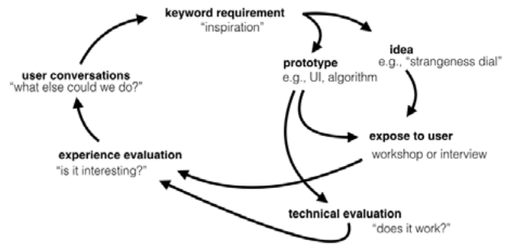


Figure 2. User involvement flow as a circular process.

As concrete examples, from our open-ended interview sessions, a number of ideas and requirements have emerged addressing the areas of structuring audio repositories and describing and finding sounds, embodiment and physical devices, and the role of the (collaborating) machine in music creation. In the following sections we will lead with example statements from our expert users to demonstrate how these findings are informing the technical research in the project and to illustrate the circular flow of the development process depicted in fig. 2. More details on the studies and general considerations of the methodological approach can be found in [4, 5, 6, 7].

3. MIR FOR MUSIC PRODUCTION

Music information retrieval plays a central role in the project. The goal of the research is to develop high-performance and low-complexity methods for music and audio analysis that allow for extraction of musical knowledge in order to drive intelligent composition agents and visualizations (cf. section 4). The following user quotes demonstrate a need for accurate music analysis methods.

> *"Onset detection, beat detection, tempo detection and harmony detection is pretty much all we need. [...] Being able to pick out a small musical phrase of any kind in a big bunch of noises could be extremely helpful for people like me. Instead of spending nights equalizing something to get out a small musical idea."* [Tok003]

> *"...if you had technology that could tag all your drum samples, that this one is like dirty or distorted, 43Hz is the dominant frequency..."* [Tok007]

Inspired by these and other user statements, we developed new and improved MIR algorithms for onset detection, beat and downbeat tracking [8, 9, 10, 11], tempo estimation [12, 13], key detection [14], chord extraction, melody extraction, style description and classification, instrument classification, drum sample description and recommendation, and drum transcription [15] for electronic dance music (EDM).

Our methods for onset detection, beat tracking, and tempo estimation [4] have successfully competed in the scientific MIREX evaluation campaign and yielded the top ranks in their respective tasks in two consecutive years. [5] [6] Furthermore, steps towards the optimization of algorithms for mobile platforms have been undertaken by establishing an audio analysis and benchmarking framework for the iOS mobile platform and a real-time-capable analysis library [7] for use in Pure Data and Max/MSP environments, both based on the Essentia audio analysis library. [8] The released libraries are not only of interest to researchers but also address music hackers and developers who often are music practitioners themselves. In addition to signal-based approaches to music analysis, we also investigate the potential of online resources to provide semantic information on music and music styles [17]. Developed software libraries and tools are made available via the GiantSteps GitHub account. [9]

To facilitate MIR research in these areas also outside the consortium, two test collections for tempo and key detection in EDM were created and released [18]. [10] The GiantSteps Key data set has already been adopted as an evaluation data set for the MIREX 2015 key detection task.

[4] made available via https://github.com/CPJKU/madmom

[5] http://www.music-ir.org/mirex/wiki/2014:MIREX2014_Results

[6] http://www.music-ir.org/mirex/wiki/2015:MIREX2015_Results

4. EXPERT AGENTS AND NEW INTERFACES

The musical knowledge extracted through MIR methods is used to inform supportive and inspirational music expert agents as well as enable new visualisations and interfaces. While users generally welcome the possibility of compositional support by intelligent systems, we found that this is a sensitive matter as it can not only disturb the creative process but also challenge the artistic identity of the user.

> *"It can turn to be pretty invasive pretty fast, and, like, annoying, if it doesn't work properly."* [Ber002]

> *"I am happy for it to make suggestions, especially if I can ignore them."* [Tok007]

> *"I'm sceptical about introducing, you know, stuff like melody into it, like, here's a suggested kind of thing which fits nicely the two or three patterns you already got in there, then you are really kind of like creating melodies for them, then it's like (laughs), then it's really like, you know, who is the composer of this?"* [Ber003]

Thus, we have to ensure that these systems are present when needed, but do not take over or inhibit the creative process. So far, the expert agents developed were encapsulated in designated UI modules that can be invoked when seeking inspiration but otherwise do not invade the existing workflow. Developed suggestion systems are concerned with rhythmic variations, tonality-aware scale restrictions, concatenative sound generation based on timbre, or arpeggiations, among others.

Due to the importance of rhythm and drum tracks in many contemporary electronic dance genres, quite some effort was devoted to rhythm pattern variation and generation. The goal of this research is to develop algorithms to recommend variations of a rhythm pattern to a user in an interactive way for usage in live situations and composition. So far, three different approaches were designed and compared, namely pattern variation based on database-retrieval, restricted Boltzmann machines [19], and genetic algorithms [20]. Fig. 3 shows the interface consisting of a simple drum pattern grid editor and a dial for effortless variation, which was received very positively due to its simplicity and creative output.

[7] http://mtg.upf.edu/technologies/EssentiaRT~

[8] http://essentia.upf.edu

[9] https://github.com/GiantSteps

[10] The test collections together with benchmarking results comparing academic algorithms and commercial products can be found at http://www.cp.jku.at/datasets/giantsteps/.

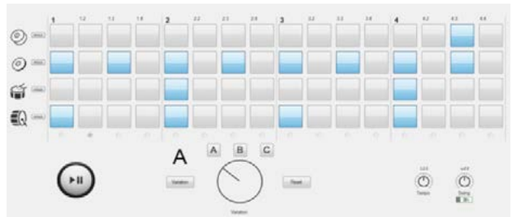


Figure 3. Intelligent user interface for rhythm variation, controllable via hardware interfaces and connected with digital audio workstation through MIDI in and out.

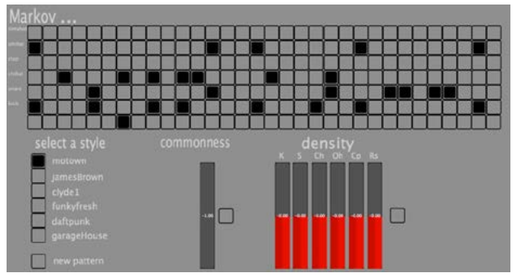


Figure 4. Drumming with style/Markov Drums, running as a Pure Data application inside a VST container.

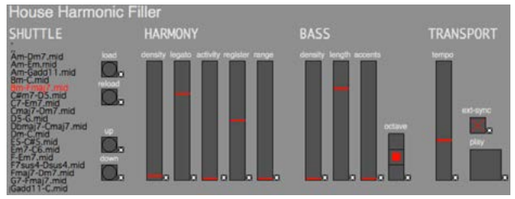


Figure 5. House Harmonic Filler, implemented as a Pure Data application with MIDI-learn control capabilities and master/slave synchronization with external MIDI clocks.

The prototype shown in fig. 4, is an interactive drum pattern generator based on Markov chains incorporating findings on rhythm similarity through user studies [21, 22, 23]. It addresses rhythm variation from a performance perspective, allowing continuous variations to be controlled by the performer on the basis of high-level musical parameters such as density, syncopation, commonness, amount and rate of variation, while maintaining the drumming style loaded or predefined by the user.

Other prototypes, aim at chord variation in live performance contexts of, currently, House music (see fig. 5), visual browsing interfaces for concatenative synthesis drum generation (see fig. 6), or integrate multiple prototypes to control several facets simultaneously [24].

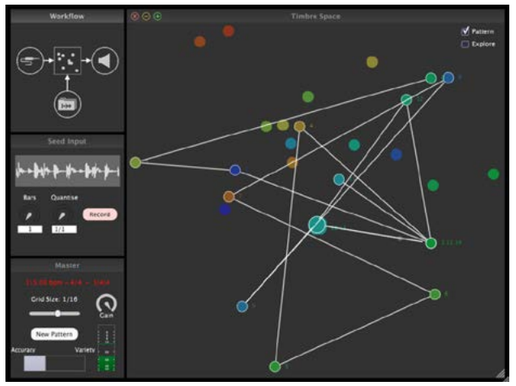


Figure 6. RhythmCAT, a VST-based software instrument that generates new patterns through granular synthesis of sounds clustered in 2D based on timbral similarity.

The integration of these agents in the workflow of music creators is inherently tied to the development of suitable interfaces for either existing desktop-based production and performance suites (i.e., digital audio workstations, such as Apple's Logic, Ableton's Live, Avid's ProTools, Steinberg's Cubase, or NI's Maschine), tangible and/or tabletop user interfaces like the Reactable [25], or smaller multi-touch interfaces of affordable portable devices such as tablets and smartphones. For instance, a developed *automatic tonalizer* expert agent integrates with the Reactable by displaying a virtual keyboard that is restricted to notes that match the scale of sample objects positioned on the table. The impact of the *intelligent arpeggiator, scaler, and chorder* agent can be controlled by hardware dials on a new hardware interface (cf. section 5). Other interface developments relate to the collaborative control of multi-dimensional parameter spaces, leading to intuitive, expressive and tangible input modalities [26, 27].

5. PRODUCT INTEGRATION

Apart from inclusion into publicly accessible developer libraries (cf. section 3), the maturity of the technical developments in the project have allowed us to integrate some of the project's outcomes into market-ready commercial products already. For instance, the Intelligent Arpeggiator, Scale, and Chord Engine has been integrated and released by NI into the Komplete Kontrol Plugin, a plugin shell that is shipped with the Komplete Keyboard (fig. 7) for seamlessly browsing through Komplete instruments, and the iMaschine 2 app for iOS (fig. 8) which was the no.1 app on the US iTunes store for several weeks end of 2015.

The same features were also released as a free update to existing Maschine customers with the free Maschine 2.2 "Melody" update in Nov. 2014, reaching +100k users. The developed Automatic Tonalizer has been integrated by Reactable Systems and will be contained in a future release (see fig. 9). This integration effort will intensify in the third and final year of the project, as more ideas and prototypes mature.

Figure 7. The Native Instruments Kontrol keyboard released in 2015 containing the Intelligent Arpeggiator, Scale, and Chord Engine developed within GiantSteps.

Figure 8. The Native Instruments iMaschine 2 app released in 2015 containing GiantSteps technology.

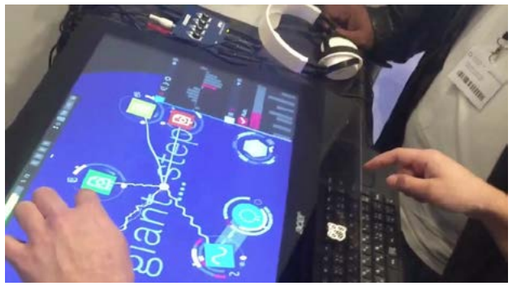


Figure 9. The Reactable Automatic Tonalizer being showcased at Musikmesse 2015.

6. CONCLUSIONS AND OUTLOOK

The consortium and orientation of GiantSteps allow for a genuinely target-user-focused MIR and HCI research approach. This unusual combination of disciplines makes it possible for user's requests and desires to be present in the earliest stages of the MIR algorithm design, a process users are otherwise often excluded from.

While the first two years have primarily been concerned with the *extraction* of musical knowledge by means of MIR technology and the *application* of this knowledge in music expert agents, the third year will have a focus on *interacting* with this knowledge, thus stressing the need for intuitive interfaces as well as for possibilities for collaboration — both, with other musicians and intelligent machines.

Through this process the underlying question for the user remains: Which technology would you want, if you could have anything at all? In practice, as the process moves on, this is refined to questions like: When is an algorithm good enough? How can you annotate or "mark" musical fragments, so that they remain available to you (see also [28])? Can you imagine a system that is able to make valuable suggestions in real time? Could these suggestions also serve as push-back and creative obstructions? and finally: What will it mean for your music, if it works?

Throughout our conversations with users, there are strong desires for improved retrieval mechanisms and inspirational systems that help exploring "the other," e.g., through non-obvious, serendipitous recommendations.

> *"...what takes me really long time is organizing my music library for DJing. [...] it could be something like Google image search for example."* [Tok011]
>
> *"Because we usually have to browse really huge libraries [...] that most of the time are not really well organized."* [Tok003]

In relation to supportive and recommendation systems, i.e., to the question "how do we want the computer to 'help' us in our creative work process?", beside issues of artistic control and the fear of making predictable sounds, it becomes apparent that the desired features of recommenders in a creative context go beyond the query-by-example-centered paradigm of finding similar items and even also beyond the goal of serendipitous suggestions, cf. [29].

> *"What I would probably rather want it would do is make it complex in a way that I appreciate, like I would be more interested in something that made me sound like the opposite of me ... but within the boundaries of what I like, because that's useful. Cause I can't do that on my own, it's like having a band mate basically."* [Tok007]

So the desired functionality of the machine is to provide an alter-ego of sorts, which provides the artist with opposite suggestions, that still reside within the artist's idea of his own personal style. This can be related to the artistic strategy of "obstruction" to assess the quality of a piece in the making, by changing the perception of the freshly edited music through changes in acoustics and hardware to render the piece "strange" [30].

This must of course be done with a strong consideration of how each musician's notion of "strange" depends on personality, emotions, preferences, and style of music, cf. [31].

> *"No, it should be strange in that way, and then continue on in a different direction. That's the thing about strange, that there's so many variations of strange. There's the small, there's the big, there's the left, there's the right, up and down."* [Strb006]

In addition to the more concrete steps of elaborating on interaction with musical knowledge, we will keep exploring these open questions. Throughout this process we are determined to not only investigate if these ideas work, but maybe more importantly, if they are interesting and productive as interfaces for creative expression in digital sound.

Acknowledgments

This work is supported by the European Union's seventh Framework Programme FP7 / 2007-2013 for research, technological development and demonstration under grant agreement no. 610591 (GiantSteps).

7. REFERENCES

[1] P. Knees, K. Andersen, S. Jordà, M. Hlatky, G. Geiger, W. Gaebele, and R. Kaurson, "GiantSteps – Progress Towards Developing Intelligent and Collaborative Interfaces for Music Production and Performance," in *2015 IEEE International Conference on Multimedia & Expo (ICME) Workshop Proceedings*, 2015.

[2] K. Andersen and D. Gibson, "The Instrument as the Source of new in new Music," in *Proceedings of the 2nd Biennial Research Through Design Conference (RTD)*, Cambridge, UK, 2015.

[3] K. Andersen, "Using Props to Explore Design Futures: Making New Instruments," in *Proceedings of the ACM CHI 2014 workshop on Alternate Endings: Using Fiction to Explore Design Futures*, Toronto, Canada, 2014.

[4] K. Andersen and F. Grote, "GiantSteps: Semi-Structured Conversations with Musicians," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA)*, Seoul, Republic of Korea, 2015.

[5] F. Grote, K. Andersen, and P. Knees, "Collaborating with Intelligent Machines: Interfaces for Creative Sound," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA)*, Seoul, Republic of Korea, 2015.

[6] F. Grote, "Jamming with Machines: Social Technologies in Musical Creativity," in *Proceedings of the 8th midterm Conference of the European Research Network Sociology of the Arts*, Cluj, Romania, 2014.

[7] ——, "The Music of Machines – Investigating Culture and Technology in Musical Creativity," in *Proceedings of the XIII. Conference on Culture and Computer Science (KuI)*, Berlin, Germany, 2015.

[8] S. Böck and G. Widmer, "A Multi-Model Approach to Beat Tracking Considering Heterogeneous Music Styles," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014.

[9] M. Davies and S. Böck, "Evaluating the Evaluation Measures for Beat Tracking," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014.

[10] F. Korzeniowski, S. Böck, and G. Widmer, "Probabilistic Extraction of Beat Positions From Neural Network Activations," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014.

[11] F. Krebs, S. Böck, and G. Widmer, "An efficient state space model for joint tempo and meter tracking," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, 2015.

[12] S. Böck, F. Krebs, and G. Widmer, "Accurate Tempo Estimation based on Recurrent Neural Networks and Resonating Comb Filters," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, 2015.

[13] F. Hörschläger, R. Vogl, S. Böck, and P. Knees, "Addressing Tempo Estimation Octave Errors in Electronic Music by Incorporating Style Information Extracted from Wikipedia," in *Proceedings of the 12th Sound and Music Conference (SMC)*, Maynooth, Ireland, 2015.

[14] Á. Faraldo, E. Gómez, S. Jordà, and P. Herrera, "Key Estimation In Electronic Dance Music," in *Proceedings of the 38th European Conference on Information Retrieval (ECIR)*, Padua, Italy, 2016.

[15] M. Leimeister, "Feature Learning for Classifying Drum Components from Nonnegative Matrix Factorization," in *Proceedings of the 138th International Audio Engineering Society Convention (AES)*, Warsaw, Poland, 2015.

[16] B. Lehner, G. Widmer, and S. Böck, "A low-latency, real-time-capable singing voice detection method with LSTM recurrent neural networks," in *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, 2015.

[17] P. Knees, "The Use of Social Media for Music Analysis and Creation Within the GiantSteps Project," in *Proceedings of the First International Workshop on Social Media Retrieval and Analysis (SoMeRA)*, Gold Coast, Queensland, Australia, 2014.

[18] P. Knees, Á. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. Le Goff, "Two Data Sets for Tempo Estimation and Key Detection in Electronic Dance Music Annotated from User Corrections," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, 2015.

[19] R. Vogl and P. Knees, "An Intelligent Musical Rhythm Variation Interface," in *Proceedings of the 21st International Conference on Intelligent User Interfaces (IUI)*, Sonoma, CA, USA, 2016.

[20] C. Ó Nuanáin, P. Herrera, and S. Jordà, "Target-Based Rhythmic Pattern Generation and Variation with Genetic Algorithms," in *Proceedings of the 12th Sound and Music Conference (SMC)*, Maynooth, Ireland, 2015.

[21] D. Gómez-Marín, S. Jordà, and P. Herrera, "Strictly Rhythm: Exploring the effects of identical regions and meter induction in rhythmic similarity perception," in *Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Plymouth, UK, 2015.

[22] ——, "Evaluating rhythm similarity distances: the effect of inducing the beat," in *Proceedings of the 15th Rhythm Production and Perception Workshop (RPPW)*, Amsterdam, the Netherlands, 2015.

[23] ——, "PAD and SAD: Two Awareness-Weighted Rhythmic Similarity Distances," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, 2015.

[24] Á. Faraldo, C. Ó Nuanáin, D. Gómez-Marín, P. Herrera, and S. Jordà, "Making Electronic Music with Expert Musical Agents," in *ISMIR 2015: 16th International Society for Music Information Retrieval Conference Late Breaking/Demo Session*, Málaga, Spain, 2015.

[25] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina, "The reacTable," in *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, Spain, 2005.

[26] K. Gohlke, M. Hlatky, and B. de Jong, "Physical Construction Toys for Rapid Sketching of Tangible User Interfaces," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI)*, Stanford, CA, USA, 2015.

[27] C. Ó Nuanáin and L. Ó Sullivan, "Real-time Algorithmic Composition with a Tabletop Musical Interface: A First Prototype and Performance," in *Proceedings of the 9th Audio Mostly: A Conference on Interaction With Sound (AM)*, Aalborg, Denmark, 2014.

[28] P. Knees and K. Andersen, "Searching for Audio by Sketching Mental Images of Sound – A Brave New Idea for Audio Retrieval in Creative Music Production," in *Proceedings of the 6th ACM International Conference on Multimedia Retrieval (ICMR)*, New York, NY, USA, 2016.

[29] P. Knees, K. Andersen, and M. Tkalčič, "'I'd like it to do the opposite': Music-Making Between Recommendation and Obstruction," in *Proceedings of the 2nd International Workshop on Decision Making and Recommender Systems (DMRS)*, Bolzano, Italy, 2015.

[30] K. Andersen and P. Knees, "The Dial: Exploring Computational Strangeness," in *Proceedings of the 34th Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA)*, San Jose, CA, USA, 2016.

[31] B. Ferwerda, "The Soundtrack of My Life: Adjusting the Emotion of Music," in *CHI Workshop on Collaborating with Intelligent Machines: Interfaces for Creative Sound*, Seoul, Republic of Korea, 2015.

A Supervised Approach for Rhythm Transcription Based on Tree Series Enumeration

Adrien Ycart
Sorbonne Universités
STMS (IRCAM-CNRS-UPMC)
Paris, France
adrien.ycart@ircam.fr

Florent Jacquemard
INRIA – Sorbonne Universités
STMS (IRCAM-CNRS-UPMC)
Paris, France
florent.jacquemard@inria.fr

Jean Bresson
Sorbonne Universités
STMS (IRCAM-CNRS-UPMC)
Paris, France.
jean.bresson@ircam.fr

Sławek Staworko
University of Edinburgh
Scotland
slawomir.staworko@inria.fr

ABSTRACT

We present a rhythm transcription system integrated in the computer-assisted composition environment OpenMusic. Rhythm transcription consists in translating a series of dated events into traditional music notation's pulsed and structured representation. As transcription is equivocal, our system favors interactions with the user to reach a satisfactory compromise between various criteria, in particular the precision of the transcription and the readability of the output score. It is based on a uniform approach, using a hierarchical representation of duration notation in the form of rhythm trees, and an efficient dynamic-programming algorithm that lazily evaluates the transcription solutions. It is run through a dedicated user interface allowing to interactively explore the solution set, visualize the solutions and locally edit them.

1. INTRODUCTION

We call rhythm transcription the act of converting a temporal stream such as the onsets in a sequence of notes into a musical score in Western notation. The note series can come from a musician's performance, or can be generated by an algorithm, for instance in a computer-assisted composition (CAC) environment such as OpenMusic [1]. In this article, we will particularly focus on the latter case.

Rhythm transcription is a long-discussed computer-music challenge [2], which can be divided into several sub-tasks (beat tracking, tempo/meter estimation, etc.) that are often considered as Music Information Retrieval (MIR) problems on their own [3, 4].

In traditional music notation, durations are expressed as fractions of a unit (*beat*) given by the *tempo*. The durations in physical time (in seconds) thus need to be converted in musical time, which requires a tempo (in beats per minute) to be inferred. The duration values have to belong to the small set defined by successive divisions of the beat (eighth notes, sixteenth notes, *etc*). The input durations thus have to be approximated (once converted into musical time) by admissible note values. We call this task *rhythm quantization*. Transcription can also be made easier by a first *segmentation* step, cutting the input stream into smaller units (if possible, of constant tempo) that are easier to analyze.

One of the difficulties of rhythm transcription stems from the coupling between tempo estimation and quantization. On the one hand, the durations cannot be quantized without knowing the tempo, and on the other hand, the quality of the transcription can only be assessed after obtaining the result of quantization. The situation is thus a chicken-and-egg problem [5].

Apart from that problem, rhythm quantization itself is difficult as the solution is not unequivocal: for a given input series of notes, several notations are admissible, and they can be ranked according to many different criteria. One of the criteria is the *precision* of the approximation, *i.e.* how close the output is to the input in terms of timing. Another important criterion is the *complexity* of the notation, *i.e.* how easy it is to read. These two criteria are often contradictory (cf. figure 1) : in general, the more precise the notation is, the more difficult it is to read. Thus, to yield good results, quantization must be a compromise between various criteria.

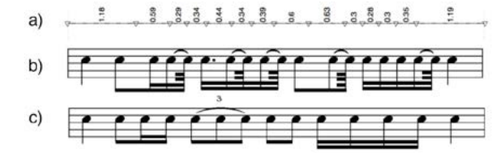


Figure 1. Figure taken from [6] : a) Input sequence. b) A precise but complex notation of the input sequence. c) Another notation of the same sequence, less precise but less complex.

Moreover, the same series of durations can be represented by various note values, as shown in Figure 2. Even if they represent the same durations, some of those transcriptions can be more readable than others. They can also have different musical meanings, and be interpreted differently.

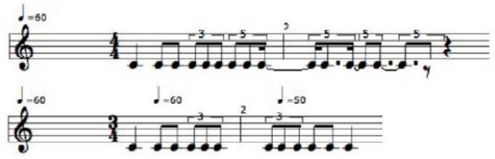


Figure 2. Two equivalent notations of the same series of durations, the second one being more readable.

All these ambiguities make rhythm quantization a hard problem, and developing an fully automatic system that compromises between precision and complexity, all the while respecting, in the case of CAC, the message the composer wants to convey, is not realistic. Moreover, a single-solution approach, returning the *optimal* transcription may be unsatisfactory in many cases. Indeed, there is no ideal compromise between the criteria.

In this article, we present a multi-criteria enumeration approach to transcription, integrated in OpenMusic. Our aim is to enumerate the various possible transcriptions, from best to worst, according to a given set of criteria. This approach is different from a single-solution approach, as we study supervised, interactive frameworks, where the user guides the algorithm throughout the process to converge to a solution. Besides, our approach allows an original coupling of the tempo estimation and quantization tasks.

The first step is the construction of a structure to guide the enumeration according to a *schema* given by the user. Intuitively, the schema describes how beats can be cut, *i.e.* what durations are admissible and in which order, thus it is a formal language. Then we run a dynamic-programming algorithm to enumerate lazily all the solutions given by the various divisions of the beat allowed by the schema, ranked according to quality criteria. A user interface allows to prepare the data, select and edit among the results obtained by the algorithm.

Our system is intended to be used in the context of CAC. Thus, it is primarily designed to quantize inputs for which there is no pre-existing score, because the score being composed, as opposed to inputs which are performances of an existing piece. We assume nothing about how the input was generated, our goal is to find the notation that best represents it. In this way, our system is style-agnostic. In particular, it does not use performance models to make up for performance-related imprecisions (such as swing in jazz).

After a brief state of the art, we define in section 3 the schema used, along with the quality criteria and the enumeration algorithm. In section 4, we describe the transcription scenarios made possible by our tools and its user interface. In section 5, we compare our system to existing solutions, and discuss the results.

2. STATE OF THE ART

Quantization is an old and complex problem. Many quantization systems exist on the market, integrated in score editors or digital audio workstations (typically, to visualize MIDI data in music sheets). But in most cases, the results are unsatisfactory when the input sequences are too irregular or complex. Besides, the user has very few parameters to influence the result, apart from manually editing it after transcription.

Some systems (in particular the one described in [7]) are based on ratios between successive durations, that must be the ratio of the smallest possible integers. Ali Cemgil et al. proposed a Bayesian model for rhythm transcription [8], in which a performance model with Gaussian noise is used. OpenMusic's current quantization tool, *omquantify* [9], aligns input note onsets on uniform grids in each beat, and chooses the one that gives the best compromise between precision and complexity.

These systems have interesting properties, but they all suggest a unique solution: if the result is unsatisfactory, the algorithm has to be re-run with different parameters.

The OMKant [9] library (not available in recent OpenMusic versions) proposed a semi-supervised approach for segmentation and rhythm transcription. The user could segment the input stream manually or automatically with various algorithms. A variable tempo estimation algorithm placed the beats, and the quantization step was done by *omquantify*. A user interface allowed to set and visualize various parameters (such as the marks used for segmentation), and to choose between the various tempo values suggested by the algorithm.

A framework for score segmentation and analysis in OpenMusic was more recently proposed in [10], which can be used for rhythm transcription as well. This framework allows to segment a note stream to transcribe it with *omquantify* using a different set of parameters on each segment. Such approach provides the user with better control on the final result, and allows more flexibility in specifying the parameters.

3. QUANTIZATION ALGORITHM

We first present the problem we want to address and the tools we use for this purpose. Quantization consists in aligning some input points to *grids* of authorized time values. We start by defining formally this problem and then present the formalisms that we shall use for the representation of grids and the output of the problem.

3.1 The Quantization Problem

We consider an *input* flow of monophonic (non-overlapping) notes and rests, represented by the increasing sequence of their respective starting dates $\boldsymbol{x} = (x_1, \ldots, x_n)$ in an interval $I_0 = [x_0, x'_0[$. Intuitively, for $i \geq 1$, the ith event (note or rest) will start at the date x_i and, terminate at x_{i+1} (the starting date of the next event), if $i < n$, or terminate at x'_0 if $i = n$. [1]

As an additional input, we also consider a set G of increasing sequences $(z_0, \ldots, z_m)$ of dates in the same interval $[x_0, x'_0[$, such that $m \geq 1$, $z_0 = x_0$ and $z_m = x'_0$. Each sequence in G is called a *grid*, and every interval $[z_i, z_{i+1}[$ between two successive points is called a *segment* of the grid. The grid is called *trivial* if $m = 1$. The exact representation of sets of grids is described in Section 3.3.

A quantization *output* is another increasing sequence of dates $\boldsymbol{y} = (y_1, \ldots, y_n)$ in the interval $[x_0, x'_0[$, such that there exists a grid $(z_0, \ldots, z_m) \in G$ and y_i belongs to $\{z_0, \ldots, z_m\}$ for each $1 \leq i \leq n$. It will be presented as a *rhythm tree*, as explained in Section 3.4.

Our goal is to produce the possible outputs given $\boldsymbol{x} = (x_1, \ldots, x_n)$ and G, enumerated according to a fixed weight function which associates a real value to every couple (input ; output) (see Section 3.5).

[1] If we want to concatenate $\boldsymbol{x}$ to another input $\boldsymbol{x'}$ in $[x'_0, x''_0[$, then the termination of x_n is set to the first date in $\boldsymbol{x'}$ – these details are left out of this paper.

3.2 Uniform Grids

Most quantization algorithms consider a finite set G of *uniform* grids, *i.e.* grids $(z_0, \ldots, z_m)$ whose segments all have the same length: $z_1 - z_0 = z_2 - z_1 = \ldots = z_m - z_{m-1}$. The advantage of this approach is that the number of relevant uniform grids is quite small (typically smaller than 32), hence the number of solutions defined this way is small as well and they can be enumerated in linear time. However, this approach is incomplete and may give unnecessarily complicated results (*e.g.* 7-uplets), in particular when the density of input points is not uniform – see Figure 3.

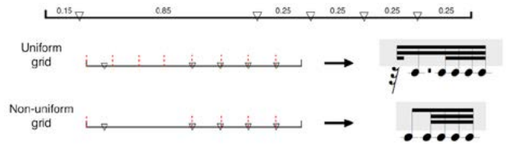


Figure 3. Quantization of an input sequence using a uniform grid (division by 8), and a non-uniform grid (division by 2 and then re-division by 4 in the second half only). The non-uniform grid gives a better result because the pitch of the grid is adapted to the density of input points.

3.3 Subdivision Schemas and Derivation Trees

For completeness purposes, we use non-uniform grids, defined by recursive subdivision of segments into equal parts. In order to define a finite set of such grids, we use a so-called *subdivision schema*, which is an acyclic context-free grammar $\mathcal{G}$ with a finite set of non-terminal symbols $\mathcal{N}$, an initial non-terminal $N_0 \in \mathcal{N}$ and a unique terminal symbol $\bullet$. The production rules are of two kind : (i) some production rules of the form: $N \to N_1 \ldots N_p$, with $N, N_1, \ldots, N_p \in \mathcal{N}$, and (ii) $\forall N \in \mathcal{N}$, $N \to \bullet$. The production rules of the latter kind will generally be ommitted.

Defining rhythms with formal grammars and derivation trees [11] (see Figure 4) is quite natural when dealing with common Western music notation, where durations are expressed as recursive divisions of a given time unit.

A derivation with $\mathcal{G}$ consist in the successive replacement of non-terminals N by the corresponding right-hand-side of production rules, starting with N_0. Intuitively, during a replacement, the non terminal N correspond to a segment of the grid (an interval), and either (i) the application of $N \to N_1 \ldots N_p$ is a division of this segment into p equal parts, or (ii) the application of $N \to \bullet$ corresponds to not dividing any further.

Every such derivation is represented by a *derivation tree* (DT) whose leaves are labeled with $\bullet$ and inner nodes are labeled with non-terminals of $\mathcal{N}$. The labels respect the production rules in a sense that if an inner node ν is labelled with N and its sons $\nu_1, \ldots, \nu_p$ are respectively labelled with $N_1, \ldots, N_p$, then there exists a production rule $N \to N_1 \ldots N_p$.

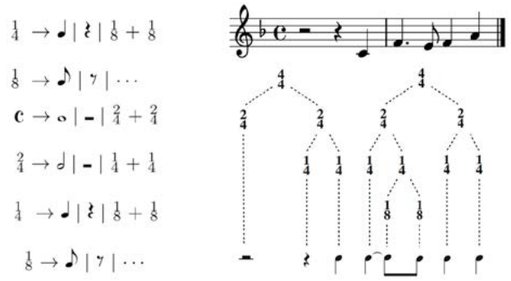


Figure 4. [11] A context-free grammar, a rhythm and the corresponding derivation tree.

Given a DT t of $\mathcal{G}$ and an initial interval $I_0 = [x_0, x'_0[$, we associate an interval to each node of t as follows:

to the root node ν_0, we associate I_0,

if ν is an inner node associated to $I = [z, z'[$ and with p sons $\nu_1, \ldots, \nu_p$, then ν_i is associated with :

$$part(I, i, p) := \left[z + \frac{(i-1)(z'-z)}{p}, z + \frac{i(z'-z)}{p}\right[.$$

The grid g_t associated to a DT t of $\mathcal{G}$ is defined by the bounds of its segments, which are the intervals associated to the leaves of t. Following the presentation in Section 3.1, we make by abuse no distinction between a schema $\mathcal{G}$, its set of DTs and the set of associated grids G.

The quantization output $\boldsymbol{y} = (y_1, \ldots, y_n)$ associated to an input $\boldsymbol{x} = (x_1, \ldots, x_n)$ and a DT t is defined as the n closest point to $x_1, \ldots, x_n$ in the grid g_t associated to t (with a default alignment to the left in case of equidistance). Given an input $\boldsymbol{x}$ and a schema $\mathcal{G}$, the set of *quantization solutions* is the set of output $\boldsymbol{y}$ associated to $\boldsymbol{x}$ and a DT of $\mathcal{G}$.

3.4 Solutions as Sets of Rhythm Trees

Searching the best alignments of the input points to some allowed grids is a classical approach to rhythm quantization. However, instead of computing all the possible grids g_t represented by $\mathcal{G}$ (as sequences of points) and the alignments of the input $\boldsymbol{x}$ to these grids, we compute only the DTs, using dynamic programming techniques.

Indeed, trees, rather than sequences, are the structures of choice for the representation of rhythms in state-of-the-art CAC environments such as OpenMusic [12] .

Our DTs (*i.e.* grids) are converted into OpenMusic rhythm trees (RT) for rendering purpose and further use by composers. The conversion is straighforward, using decoration of the leaves of DTs and tree transformation functions.

3.5 Rhythm Tree Series

In order to sort the solution set, we consider the notion of *trees series* [13], which is a function associating to each tree a weight value in a given domain—here the real numbers. In our case, the smaller the weight of a tree is, the

better the corresponding notation is. We describe below the definition of this function as a combination of several criteria.

3.5.1 Criteria and Combinations

The weight of a tree is calculated by combining several criteria, which are functions associating a real value to an input $\boldsymbol{x}$ and a DT t. We take into account a distance criterion, and a complexity criterion. They are computed recursively: the value of a criterion for a tree is evaluated using the values of criteria of his son.

The chosen *distance* criteria is defined by :

$$dist(\boldsymbol{x}, t) = \sum_{x_i \in segment(t)} |x_i - y_i|$$

for a subtree t which is a leaf, where $segment(t)$ is the set of inputs x_i contained in the interval associated to t, and $\boldsymbol{y}$ is defined as above, and

$$dist\big(a(t_1, \ldots, t_p)\big) = \sum_{i=1}^{p} dist(t_i)$$

The *complexity* criterion is defined as a combination of several sub-criteria. The first sub-criterion is related to the size of the tree and the degrees of the different nodes. It is the sum of the numbers of nodes having a certain degree, weighted by penalty coefficients. We denote by β_j the coefficient describing the complexity of the degree j, and follow the order recommended in [14] to classify arities from the simplest to the most complex: $\beta_1 < \beta_2 < \beta_4 < \beta_3 < \beta_6 < \beta_8 < \beta_5 < \beta_7$... The other sub-criterion is the number of *grace notes* present in the final notation. A grace note corresponds to the case where several entry points are aligned on the same grid point, *i.e.* $y_{i+1} = y_i$ (we recall that we consider monophonic inputs, two notes aligned to the same point do not correspond to a chord). We aim at minimizing the number of grace notes, since too many grace notes hinder readability.

If t is a leaf, then $comp(t) = g(t)$, the number of grace notes, determined by counting the number of points of the segment aligned with each of its boundaries, and

$$comp\big(a(t_1, \ldots, t_p)\big) = \beta_p + \sum_{i=1}^{p} comp(t_i)$$

The weight $w(t)$ of a tree t is a linear combination of the above criteria:

$$w(t) = \alpha.dist(t) + (1 - \alpha).comp(t)$$

The coefficient α is a parameter of the algorithm, used to adjust the relative importance of the two criteria in the final result: $\alpha = 0$ will return results favoring simplicity of notation (small rhythm trees, simple tuplets, few grace notes) at the expense of the fitness of transcription, while $\alpha = 1$ will return rhythm trees of maximum depth, often corresponding to less readable notations, but transcribing the input as accurately as possible.

3.5.2 Monotonicity

The criteria and their combination were not chosen arbitrarily, they were chosen to follow the following property of monotony for the purpose of correctness of the enumeration algorithm below.

$$\forall t = a(t_1, \ldots, t_p)\, \forall i \in [1..p]\, \forall t'_i\, w(t'_i) > w(t_i) \Rightarrow$$
$$w(a(t_1, \ldots, t_{i-1}, t'_i, t_{i-1}, \ldots, t_p) > w(a(t_1, \ldots, t_p)$$

In other words, if we replace a subtree by another sub-tree of greater weight, the weight of the super-tree will also be greater. One can check that this property holds for the functions defined as above.

3.6 Enumeration Algorithm

A given subdivision schema $\mathcal{G}$ will generaly define an exponential number of non-uniform grids, and therefore an exponential number of quantization solutions, according to the definition in Section 3.4. Hence, we would like to avoid having to compute all of them before ranking them, we want to lazily enumerate them in increasing weight. More precisely, the following dynamic programming algorithm called *k-best* [15] returns the solutions by packets of k, where k is a constant fixed by the user.

3.6.1 Enumeration Table

The k-best algorithm is based on a table T built from the schema $\mathcal{G}$. Each key of this table has the form $\langle N, I\rangle$, where N is a non-terminal of $\mathcal{G}$ and I is an interval $[z, z'[$ associated with a node labeled with N in an DT of $\mathcal{G}$. Every entry $T[N, I]$ of the table contains two lists:

***best*-list** $bests[N, I]$, containing the minimal weighted sub-DT whose root is labeled with N and associated the interval I, together with their weight and the values of $dist$ and $comp$.

***candidate*-list** $cands[N, I]$, containing sub-DT, among which the next best will be chosen (some weights might not have been evaluated yet).

The sub-DT in the above lists are not stored in-extenso but each of them is represented by a list of the form $(\langle N_1, i_1\rangle, \ldots, \langle N_p, i_p\rangle)$, called a *run*. A run in one of the two lists of $T[N, I]$ represents a sub-DT whose root is labeled with N and with p children, such that the jth child is the element number i_j in the best-list of $T[N_j, part(I, j, p)]$. The pair $\langle N_j, i_j\rangle$ is called a link to the i_j-best of $\langle N_j, part(I, j, p)\rangle$.

3.6.2 Initialization of the Table

The initialization and update of the list of candidates exploits the hypothesis of monotony of the weight function mentioned in Section 3.5.2. Indeed, this property implies that the best tree (tree of lesser weight) will be built with the best children, and therefore will be represented by a run containing only links to 1-bests. More precisely, we initialize every entry $T[N, I]$ with an empty best-list and a candidate-list containing one run $\big(\langle N_1, 1\rangle, \ldots, \langle N_p, 1\rangle\big)$ for each production rule $N \to N_1 \ldots N_p$ of $\mathcal{G}$ (division in p parts), and one empty run $()$, which corresponds to the case of a leaf in the DT (end of divisions). The weights of the runs in the initial candidate-lists is set as unknown.

As an optimization, when the intersection of the input $\boldsymbol{x}$ with I is empty, then we only put $()$ in the candidate list. Indeed, there is no need to further divide a segment containing no input points.

3.6.3 Algorithm

The enumeration algorithm is described in details in [16]. It reuses the results already computed and builds the trees in a lazy way: as indicated above, thanks to the monotony of the weight function, in order to construct the best tree, one just needs to construct the best sub-trees. The algorithm works recursively: in order to evaluate the weight of a sub-DT, it will evaluate the weight of each of its children. The main function $best(k, N, I)$ is recursive. It returns the k-best DT of $\langle N, I\rangle$, given k, N and I:

1. If the best-list of $T[N, I]$ contains k elements or more, then return the the kth run of this list, together with its associated weight and criteria values.

2. Otherwise, evaluate the weight of all the candidates in $T[N, I]$ as follows: for a run $\big(\langle N_1, i_1\rangle, \ldots, \langle N_p, i_p\rangle\big)$ in the candidate-list of $T[N, I]$ whose weight is unknown, call recursively $best\big(i_j, N_j, part(I, j, p)\big)$ for each $1 \leq j \leq p$, and then evaluate the weight and criteria values, using the values returned for its children and the equations in Section 3.5.1.

3. Once all the weights of the runs in the candidate-list of $T[N, I]$ have been evaluated, remove the run $\big(\langle N_1, i_1\rangle, \ldots, \langle N_p, i_p\rangle\big)$ of smallest weight from this list, add it to the best-list of $T[N, I]$ (together with weight and criteria values), and then add to the candidate-list the following next runs, with unknown weight:
$\big(\langle N_1, i_1 + 1\rangle, \langle N_2, i_2\rangle, \ldots, \langle N_p, i_p\rangle\big)$,
$\big(\langle N_1, i_1\rangle, \langle N_2, i_2 + 1\rangle, \ldots, \langle N_p, i_p\rangle\big), \ldots,$
$\big(\langle N_1, i_1\rangle, \ldots, \langle N_{p-1}, i_{p-1}\rangle, \langle N_p, i_p + 1\rangle\big)$.

An invariant of the algorithm is that for each entry of T, the next best tree (after the last best tree already in the best-list) is represented in the candidate list. This property stems from the monotonicity of the weight function, and ensures the completeness of the algorithm.

The algorithm is exponential in the depth of the schema (the maximal depth of a derivation tree). This value is typically 4 or 5 for obtaining usual rhythm notation. On our experiments with inputs of 10 to 20 points, the table has typically a few hundred entries (depending of the size of the schema).

3.7 Coupled Quantization and Tempo Estimation

Calling $best(k, N_0, I_0)$ will return the k^{th} best quantization of an input $\boldsymbol{x}$ in the interval I_0, according to the given schema $\mathcal{G}$. It works when the tempo is known in I_0. Estimating the tempo is a difficult problem; we propose a simple solution that gives good results in our context, by coupling tempo estimation with the quantization algorithm.

The idea is, given $\boldsymbol{x}$, $I_0 = [x_0, x'_0[$ and $\mathcal{G}$, to add to $\mathcal{G}$ a preliminary phase of division of I_0 into a certain number m of equal parts, corresponding to beats. This tempo estimation makes the hypothesis that the tempo is constant over I_0. The values for m are chosen so that the corresponding tempo is between values θ_{min} and θ_{max}, typically 40 and 200 bpm, or any range specified by the user:

$$\frac{(x'_0 - x_0) \times \theta_{min}}{60} \leq m \leq \frac{(x'_0 - x_0) \times \theta_{max}}{60} \qquad (1)$$

where x_0 and x'_0 are assumed given in physical time (seconds). This is done by constructing $\mathcal{G}'$, by addition to $\mathcal{G}$ of a new initial non-terminal N'_0 and new production rules $N'_0 \to \underbrace{N_0, \ldots, N_0}_{m}$ for all integral values of m satisfying (1). Using this new schema $\mathcal{G}'$, and adapted weight functions and criteria, the above enumeration algorithm will return the k best transcription solutions for all tempi, obtained by a coupled estimation of tempo and quantization. A variant consists in enumerating the k best quantization solutions for each tempo given by the possible values of m.

4. INTEGRATION IN OPENMUSIC

The algorithm presented in the previous section has been implemented as an OpenMusic library. A user interface (Figure 5) has also been developed to monitor and make use of the results of this algorithm.

The user first loads as input stream $\boldsymbol{x}$ a CHORD-SEQ object, which is the analogous of a MIDI file in terms of time representation (the start and end dates of notes are expressed in milliseconds). The rhythmic transcription is then performed in 4 steps.

1. Segmentation: The user segments the input stream of notes $\boldsymbol{x}$ in the top left panel. Typically, the length of a segment should be in the order of one bar. Moreover, these segments should preferably correspond to regions where the tempo is constant (see Section 3.7).[2] The user can also specify different parameters for each segment, such as the subdivision schemas and tempi bounds.

2. Quantization: The algorithm is run on each segment independently, to compute the k best solutions.

3. Choice of a solution: The user sees the k best transcriptions in the right panel, and selects, for each segment, one of them. The $dist$ values for each transcription are indicated. The selected transcriptions are then concatenated and displayed in the bottom left panel.

4. Edition of the solution: The user can edit the chosen solution, with the content of the table T used by the quantization algorithm. When he/she selects one region corresponding to a sub-tree in the transcription, he can visualize the *best*-list for this region and choose in the list an alternate solution for it to be replaced in the final score. The user can also estimate the $dist$ value for each sub-tree via a color code. At any time he/she can request to extend the list with the following k best solutions for this sub-tree.[3]

5. EVALUATION

Automated evaluation of the system we presented is difficult, as there is no unique objective criteria to evaluate a "good" transcription. One method could be to quantize performances of given pieces and check if the system outputs the original score. This method has two disadvantages. First, our approach is interactive: the right result will not necessarily be the first one, but it might be among the transcriptions given by the algorithm. If not, we could easily obtain it by editing the solution. Rather than counting the number of matching first result, we should count the number of operations to obtain the targeted score. Moreover, our system was not designed for performance transcription. The constant-tempo hypothesis is not adapted in this case: a variable-tempo model would yield better

[2] An automatic constant-tempo regions segmentation algorithm is currently under development.

[3] A video demonstration of the system is available at `http://repmus.ircam.fr/cao/rhythm/`.

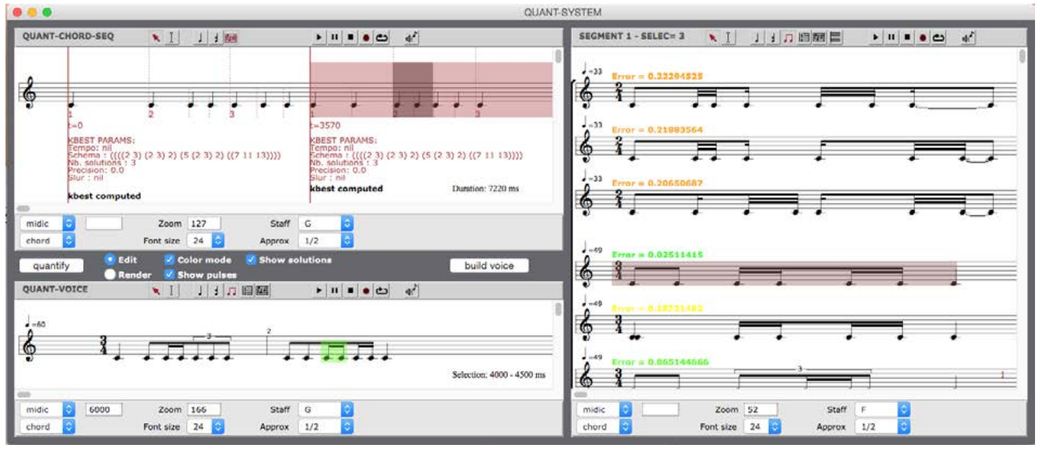


Figure 5. Screenshot of the transcription system's user interface in OpenMusic

results and thus, our system's results will not be representative of its quality. We are currently favoring test sessions with composers, in order to assess the relevance and user-friendliness of our system.

For illustration purposes, let us consider the input series of durations is the rhythm given in Figure 1c), to which we added a uniform noise between -75 and 75ms. We transcribed it with *omquantify*, with the score editor Sibelius 6 via the MIDI import function, and with our system. The results are shown in Figure 6.

Sibelius outputs a result that is easily readable, but quite far from the input score. There is no tempo estimation, the tempo of the MIDI file is directly used to quantize. However, in many cases, when tempo is not known, the MIDI file's tempo is set to default (here, 60 beats per minute), which leads to an incorrect transcription. Moreover, many durations are badly approximated: the third and fourth notes are supposed to be of same duration, as well as the 4 sixteenth notes at the end.

omquantify outputs good results, but to obtain them, the user has to input the right tempo and the right signature (3/4) as parameters. There is no tempo estimation in *omquantify*, and finding the right tempo can be very tricky, as discussed in Section 1. Otherwise, with default parameters (tempo = 60, signature = 4/4), the results might be exact, but it is inexploitable, as it is too complicated.

With default parameters ($\alpha = 0.5$), and with an adequate segmentation, our system gives good results. The estimated tempo is very close to the original (though not exact in the second bar), and the right signature is found automatically. The note values are also very close to the original, except for the triplet. Nevertheless, it should be underlined that the solution shown here is only the first proposition made by the algorithm. The goal transcription can in fact be obtained by choosing the third proposition for the first bar, and the first for the second bar. Besides, by changing the α parameter, we can get the goal rhythm as the first proposition made by the algorithm.

These good results rely on a good preliminary segmentation : we cut the input stream according to the original

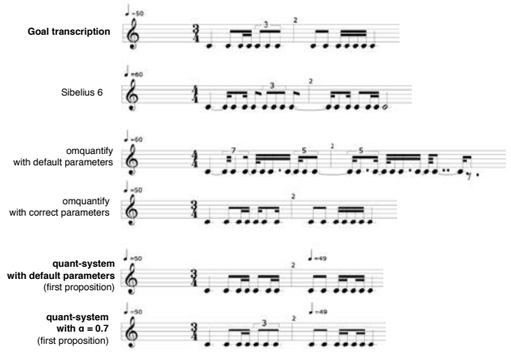


Figure 6. Results of quantization with various systems. Sibelius gives a result quite far from the input. *omquantify* gives good results, but the user has to find the right set of parameters. Our system works well with default parameters (the first proposition is not exact, the third is), and can give even better results by adjusting them.

bars. As our system considers each segment as a bar, we wouldn't have had the same result with a different segmentation. Moreover, if the segmentation marks hadn't been placed on a note corresponding to a beat of the original input, the tempo estimation step might not have worked as well (hence the importance of the supervised aspect of the system). Besides, the enumeration algorithm, and thus the tempo estimation, is ran on each segment independently, which is why we have a small tempo difference between the two bars.

The results are also less satisfactory when segments are too long (more than a few bars). Indeed, a long segment entails a lot of possible tempo values (cf section 3.7), and thus, computations are longer, and it is more difficult to choose the right tempo value.

In general, the following interactive workflow has shown to be quite successful: the user ranks the solutions by complexity (with a small parameter α, see Section 3.5.1), and then refines the more difficult (dense) regions, choosing more complex and accurate alternative solutions for these regions. The advantage of this method is that since complexity and precision are antagonist criteria, the results will be ranked by complexity and also approximately by precision, and thus there is a relative regularity in the ordering of the solutions, which makes exploration easier. On the contrary, when α=0.5, some very precise and complex solutions can be ranked close to imprecise and simple solutions, as they may have similar weights.

6. CONCLUSION

We have presented a new system for rhythmic transcription. The system ranks the transcription solutions according to their distance to input and complexity, and enumerates them with a lazy algorithm. An interface allows the user to choose from the available transcriptions and edit it in a semi-supervised workflow. At the time of this writing this tool is being tested with composers.

Finding relevant parameters (α, and β_j's) is a sensitive problem in our approach. One could use for this purpose a corpus of pairs performance/scores, such as *e.g.* the Kostka-Payne corpus [17], in order to learn parameter values that maximize fitness to some extend (number of correct transcriptions in first rank, number of correct transcriptions in the first n solutions...). We could get around the problem of variable-tempo performances by segmenting the input stream in beats, in order to focus on quantization only.

The choices made by the user could also be used to learn some user preferences and improve the results of transcription, as it was proposed in [18]. For example, the user could specify the solution he wants to keep, and the solutions he doesn't want to see again. This information can then be used to adapt the *comp* function so that the kept solutions have a smaller weight, and the unwanted ones have a higher weight.

Finally, an alternative approach consist in considering a vectorial weight domain, partially ordered by component-wise comparison, and enumerate the Pareto front (*aka skyline* [19]).

Acknowledgments

This project was supported by the project with reference ANR-13-JS02-0004 and IRCAM's "rhythm" UPI.

We wish to thank the composers who help us in the design and evaluation of our tool, in particular Julia Blondeau, Karim Haddad, Daniele Ghisi and Mikhail Malt.

7. REFERENCES

[1] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue, "Computer-assisted composition at IRCAM: From PatchWork to OpenMusic," *Computer Music Journal*, vol. 23, no. 3, pp. 59–72, 1999.

[2] M. Piszczalski and B. A. Galler, "Automatic music transcription," *Computer Music Journal*, vol. 1, no. 4, pp. 24–31, 1977.

[3] M. A. Alonso, G. Richard, and B. David, "Tempo and Beat Estimation of Musical Signals," in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, Barcelona, Spain, 2004.

[4] A. Klapuri *et al.*, "Musical meter estimation and music transcription," in *Cambridge Music Processing Colloquium*, 2003, pp. 40–45.

[5] A. T. Cemgil, "Bayesian Music Transcription," Ph.D. dissertation, Radboud Universiteit Nijmegen, 2004.

[6] P. Desain and H. Honing, "The quantization of musical time: A connectionist approach," *Computer Music Journal*, vol. 13, no. 3, pp. 56–66, 1989.

[7] D. Murphy, "Quantization revisited: a mathematical and computational model," *Journal of Mathematics and Music*, vol. 5, no. 1, pp. 21–34, 2011.

[8] A. T. Cemgil, P. Desain, and B. Kappen, "Rhythm quantization for transcription," *Computer Music Journal*, vol. 24, no. 2, pp. 60–76, 2000.

[9] B. Meudic, "Détermination automatique de la pulsation, de la métrique et des motifs musicaux dans des interprétations à tempo variable d'oeuvres polyphoniques," Ph.D. dissertation, UPMC - Paris 6, 2004.

[10] J. Bresson and C. Pérez Sancho, "New framework for score segmentation and analysis in OpenMusic," in *Proc. of the Sound and Music Computing Conf.*, Copenhagen, Denmark, 2012.

[11] C. S. Lee, "The Rhythmic Interpretation of Simple Musical Sequences: Towards a Perceptual Model," *Musical Structure and Cognition*, vol. 3, pp. 53–69, 1985.

[12] C. Agon, K. Haddad, and G. Assayag, "Representation and Rendering of Rhythm Structures," in *Proc. 2nd Int. Conf. on Web Delivering of Music*. Darmstadt, Germany: IEEE Computer Society, 2002, pp. 109–113.

[13] Z. Fülöp and H. Vogler, "Weighted Tree Automata and Tree Transducers," in *Handbook of Weighted Automata*, M. Droste, W. Kuich, and H. Volger, Eds. Springer, 2009, pp. 313–403.

[14] C. Agon, G. Assayag, J. Fineberg, and C. Rueda, "Kant: A critique of pure quantification," in *Proc. of ICMC*, Aarhus, Denmark, 1994, pp. 52–9.

[15] L. Huang and D. Chiang, "Better k-best parsing," in *Proc. of the 9th Int. Workshop on Parsing Technology*. Association for Computational Linguistics, 2005, pp. 53–64.

[16] A. Ycart, "Quantification rythmique dans OpenMusic," Master's thesis, UPMC - Paris 6, 2015.

[17] D. Temperley, "An evaluation system for metrical models," *Computer Music Journal*, vol. 28, no. 3, pp. 28–44, 2004.

[18] A. Maire, "Quantification musicale avec apprentissage sur des exemples," IRCAM, Tech. Rep., 2013.

[19] S. Borzsony, D. Kossmann, and K. Stocker, "The skyline operator," in *Proc. of the 17th Int. Conf on Data Engineering*. Heidelberg, Germany: IEEE, 2001, pp. 421–430.

```
/* Returns the k-th best run for entry T[N,I], along with its weight          */
Function best (k, N, I) :
    if length(bests[N, I]) ≥ k then
        return bests[N, I][k], w(bests[N, I][k]);
    else
        if cands[N, I] is empty then
            /* the k-best cannot be constructed, no more candidates           */
            return error
        else
            run := min(cands[N, I]) ;
            if w(run) is known then
                add run to bests[N, I];
                remove run from cands[N, I];
                add next candidates to cands[N, I] with unknown weights;
                /* cf section 3.6.3                                           */
                best(k, N, I);
                /* iterate until k bests are constructed                      */
            else
                eval(run,N, I);
                best(k, N, I);
                /* iterate until all weights are known                        */
            end
        end
    end

/* Evaluates the weight of a given run, and updates it in the cands list      */
Function eval (run,N, I) :
    if run = () then
        /* the node is a leaf, no further subdivision                         */
        compute w(run);
        /* cf section 3.5.1                                                   */
        update w(run) in cands[N, I];
    else
        /* the node has sons,recursive call to best                           */
        let run = (⟨N_1, i_1⟩, ..., ⟨N_p, i_p⟩);
        weights := (best(i_1, N_1, part(I, 1, p)), ..., best(i_p, N_p, part(I, p, p)));
        w(run) = sum of weights;
        /* cf section 3.5.1                                                   */
        update w(run) in cands[N, I];
    end
```

Algorithm 1: Pseudo-code of the enumeration algorithm

Graphical Temporal Structured Programming for Interactive Music

Jean-Michaël Celerier
LaBRI, Blue Yeti
Univ. Bordeaux, LaBRI, UMR 5800,
F-33400 Talence, France.
Blue Yeti, F-17110 France.
jcelerie@labri.fr

Myriam Desainte-Catherine
LaBRI, CNRS
Univ. Bordeaux, LaBRI, UMR 5800,
F-33400 Talence, France.
CNRS, LaBRI, UMR 5800,
F-33400 Talence, France.
INRIA, F-33400 Talence, France.
myriam@labri.fr

Jean-Michel Couturier
Blue Yeti, F-17110 France.
jmc@blueyeti.fr

ABSTRACT

The development and authoring of interactive music or applications, such as user interfaces for arts & exhibitions has traditionally been done with tools that pertain to two broad metaphors. Cue-based environments work by making groups of parameters and sending them to remote devices, while more interactive applications are generally written in generic art-oriented programming environments, such as Max/MSP, Processing or openFrameworks. In this paper, we present the current version of the i-score sequencer. It is an extensive graphical software that bridges the gap between time-based, logic-based and flow-based interactive application authoring tools. Built upon a few simple and novel primitives that give to the composer the expressive power of structured programming, i-score provides a time line adapted to the notation of parameter-oriented interactive music, and allows temporal scripting using JavaScript. We present the usage of these primitives, as well as an i-score example of work inspired from music based on polyvalent structure.

1 Introduction

This paper outlines the new capabilities in the current iteration of i-score, a free and open-source interactive scoring sequencer. It is targeted towards the composition of scores with an interactive component, that is, scores meant to be performed while maintaining an ordering or structure of the work either at the micro or macro levels. It is not restricted to musical composition but can control any kind of multi-media work.

We first expose briefly the main ideas behind interactive scores, and explain how i-score can be used as a language of the structured programming language family, targeted towards temporal compositions, in a visual time-line interface.

In previous research [1] interactive triggers were exhibited as a tool for a musician to interact with the computer following a pre-established score. Here, we show that with the introduction of loops, and the capacity to perform computations on variables in a score, interactive triggers can be used as a powerful flow control tool, which allows to express event-driven constructs, and build a notion similar to traditional programming languages procedures.

We conclude by producing an i-score example of a musical work inspired by polyvalent structure music, which can be used by composers as a starting point to work with the environment. This example contains relatively few elements, which shows the practical expressiveness of the language.

2 Existing works

The sequencer metaphor is well-known amongst audio engineers and music composers. It is generally built around tracks, which contains audio or MIDI clips, applied effects and parameter automations.

In multiple cases, it has been shown that it was possible to write more generalist multimedia time-line based sequencers, without the need to restrict oneself to audio data types. The MET++ framework [2] is an object-oriented framework tailored to build such multimedia applications. A common approach, also used in previous version of i-score, is to use constraint programming to represent relations between temporal objects [1, 3, 4]. This is inspired from Allen's relationship between temporal objects. In [5], Hirzalla shows how conditionality can be introduced between multimedia elements in a time-line to produce different outcomes.

Other approaches for interactive music are generally not based on the time-line metaphor, but more on interaction-centric applications written in patchers such Max/MSP or PureData, with an added possibility of scoring using cues. Cues are a set of parameters that are to be applied all at once, to put the application or hardware in a new state. For instance, in a single cue, the volume of a synthesizer may be fixed at the maximum value, and the lights would be shut off. However, the temporal order is then not apparent from the visual representation of the program, unless the composer takes care of maintaining it in his patch. When using text-based programming environments, such as Processing or OpenFrameworks, this may not be possible if concurrent processes must occur (e.g. a sound plays while the lights fade-in).

The syntax and graphical elements used in i-score as well

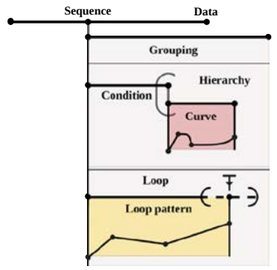


Figure 1. Screen-shot of a part of i-score, showing major elements of the formalism. The *time constraint* is the full or dashed horizontal line, the *states* are the black dots, the *time nodes* are the vertical bars, and a *time event* is shown at the right of the "Condition" text. Interactive *triggers* are black **T**'s with a downwards arrow. There are five *time processes* (capitalized): a *scenario* which is the hierarchical root of the score, another *scenario*, in the box "Hierarchy", an *automation* on a remote parameter in the "Curve" box, a *loop* in the box containing the *loop* pattern, and another *automation* that will be looped.

as the execution semantics are for the most part introduced in [6, 7], along with references to other works in the domain of interactive musical scores and presentation of the operational semantics.

The novelty of our approach lies in the introduction of graphical temporal loops, and of a computation model based on JavaScript that can be used at any point in the score. These two features, when combined, provide more expressive power to the i-score visual language, which allows for more dynamic scores.

3 Temporal structured programming

Structured programming is a paradigm which traces back to the 1960's, and was conceived at a time where the use of `GOTO` instructions was prevalent, leading to hard to read code.

The structured programming theorem [8, 9] states that any computable function can be computed without the use of `GOTO` instructions, if instead the following operations are available:

- Sequence (`A` followed by `B`),
- Conditional (`if(P) then A else B`),
- Iterative (`while(P) do A`).

Where `P` is a boolean predicate, and `A`, `B` are basic blocks. Additionally, the ability to perform computations is required in order to have a meaningful program.

To allow interactive musical scores authoring, we introduce these concepts in the time-line paradigm. A virtual machine ticks a timer and makes the time flow in the score graph. During this time, processes are computed.

Processes can be temporal or instantaneous. Temporal processes are functions of time that the composer wants to run between two points in time: *do a volume fade-in from t=10s to t=25s*.
Instantaneous processes run at a single point in time: *play a random note*.

3.1 Scenario

The *scenario* is a process and a particular setup (fig. 1) of the elements of the i-score model: *time constraint* (a span of time, which contains temporal processes), *time node* (synchronizes the ending of *time constraints* with an external event such as a note being played), *time event* (a condition to start the following *time constraints*), and *state* (contains data to send, and instantaneous processes). Time flows from left to right as in traditional sequencers. Due to the presence of interactivity, the various possibilities of execution of the score cannot be shown. Hence dashes are shown when the actual execution time is not known beforehand. For instance: *play a D minor chord until a dancer moves on stage*.

In the context of a *scenario*, as shown in [6], these primitives allow for sequencing elements, conditional branching, and interactive triggering, but are not enough for looping.

3.2 Loop

The loop is another temporal process and setup of these elements, more restrictive, and with a different execution algorithm: it is composed exclusively of two *time nodes*, two *time events*, two *states*, and a *time constraint* in-between (the loop pattern). When the second *time node* is triggered, the time flow reverts to before the execution of the first *time node*. If the composer adds an interactive trigger on any of these *time nodes*, each loop cycle may have a different duration and outcome. This is more general than loops in traditional audio sequencers, where looping only duplicates audio or MIDI data.

3.3 Communication

i-score communicates via the OSC [1] protocol, and Minuit: an OSC-based RPC [2] and discovery protocol. It maintains a tree of parameters able to mirror the object model of remote software built with Max/MSP, PureData, or any OSC-compliant environment. In the course of this paper, "device tree" refers to this tree.

3.4 Variables

Variables are based on the device tree, which acts like a global memory. They are statically typed [3]. C-like implicit conversion can take place: an integer and a floating point number will be able to be compared. There is no scoping: any process can access to any variable at any point in time. No internal allocation primitive is provided, but it can be emulated with an external software such as a PureData patch if necessary.

3.5 Authoring features

Provided temporal processes are JavaScript scripting, automations, mappings, and recordings. Execution speed can be controlled, and the score object tree can be introspected. The user interface allows for all the common and expected operations : displacement, scaling, creation, deletion, copy-paste, undo-redo... The software is based on a plug-in architecture to offer extensibility, which is how all the processes are implemented.

[1] Open Sound Control
[2] Remote Procedure Call
[3] Types are integer, boolean, floating point, impulse, string, character, or tuple.

4 Temporal design patterns

In this section, we present two design patterns that can be used for writing an interactive score. We will first showcase event-driven scores, akin to a traditional computer program executing instructions in sequence without delay, or network communication tasks. Then, we will present an example of the concept of procedure in a time-oriented model.

4.1 Event-driven design

Event-driven, or asynchronous design is a software design paradigm centered on the notion of asynchronous communication between different parts of the software. This is commonly used when doing networked operations or user interface.

In textual event-driven programming, one would write a software using callbacks, futures or reactive programming patterns [10].

One can write such event chaining easily with interactive triggers (fig. 2): B cannot happen before A if there is a *time constraint* between A and B.

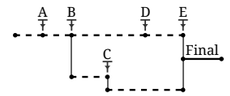


Figure 2. An example of event-driven score: if all the interactive trigger's conditions are set to true, they will trigger at each tick one after the other. Else, standard network behaviour is to be expected.

However, the execution engine will introduce a delay of one tick between each call. The tick frequency can be set to as high as one kilo-hertz. Synchronization is trivial: here, the last *time constraint* **Final**, will only be executed after all the incoming branches were executed. This allows to write a score such as: *start section B five seconds after musician 1 and 3 have stopped playing*. There is no practical limit to the amount of branches that can be synchronized in this way.

4.2 Simulating procedures

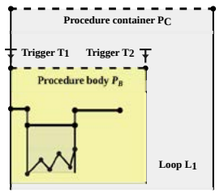


Figure 3. Implementation of a procedure in i-score.

The notion of procedure is common in imperative programming languages. It consists in an abstraction around a behaviour that can be called by name easily. However, it reduces the visual flow coherence: the definition and usage of the procedure are at different points in the score or code.

Fig. 3 gives a procedure P able to be recalled at any point in time, with a restriction due to the temporal nature of the system. It can only be called when it is not already running. This is due to the single-threaded nature of the execution engine: there is a single playhead for the score.

The procedure is built as follows:

- A *time constraint*, P_C in the root *scenario* will end on an interactive triggering set with infinite duration.
- This *time constraint* contains a *loop* L_1. The procedure is named `p` by the composer in the local tree. The interactive triggers T_1, T_2 at the beginning and end of the pattern *time constraint* are set as follows:
  - T_1: `/p/call true`.
  - T_2: `/p/call true`.

  A *state* triggered by T_1 should set the message: `/p/call false`. This causes the procedure not to loop indefinitely: it will have to be triggered manually again.
- The *loop*'s pattern P_B contains the actual procedure data, that is, the process that the composer wants to be able to call from any point in his score.

When at any point of the score, the message `/p/call true` is sent, the execution of this process overlays itself with what is currently playing. Once the procedure's execution is finished, it enters a waiting state until it is called again. This behavior is adapted to interactive arts: generally, one will want to start multiple concurrent processes (one to manage the sound, one to manage videos, one to manage lights...) at a single point in time; this method allows to implement this.

5 Musical example: polyvalent structure

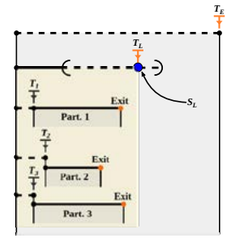


Figure 4. An example of polyvalent score in i-score

In this example (fig. 4), we present a work that is similar in structure to Karlheinz Stockhausen's *Klavierstück XI* (1956), or John Cage's *Two* (1987). This uses ideas from the two previously presented patterns. The complete work contains variables in the device tree and a temporal score. The tree is defined in fig 5.

Address	Type	Initial value
`/part/next`	integer	chosen by the composer
`/part/1/count`	integer	0
`/part/2/count`	integer	0
`/part/3/count`	integer	0
`/exit`	boolean	false

Figure 5. Tree used for the polyvalent score

`/part/next` is an address of integral type, with a default value chosen by the composer between 1, 2, 3: it will be the first played part. The score is as follows: there are multiple musical parts containing recordings of MIDI notes converted to OSC: **Part. 1, 2, 3**. These parts are contained in a *scenario*, itself contained in a *loop* that will run indefinitely. At the end of each part, there is an orange *state*

that will write a message "true" to a variable `/exit`. The pattern of the *loop* ends on an orange interactive trigger, T_L. The *loop* itself is inside a *time constraint* ended by an interactive trigger, T_E. Finally, the parts are started by interactive triggers $T_{\{1,2,3\}}$.

The conditions in the triggers are as follows:

- $T_{\{1,2,3\}}$ `/part/next == {1, 2, 3}`
- T_L `/exit == true`
- T_E $\bigvee_{i \in 1..3}$ `/part/i/count > 2`

The software contains graphical editors to set conditions easily. Finally, the blue *state* under T_L contains a JavaScript function that will draw a random number between 1 and 3, increment the count of the relevant `/part`, and write the drawn part in `/part/next` :

```
function() {
  var n = Math.round(Math.random()*2)+1;
  var root = 'local:/part/'
  return [ {
    address : root + 'next',
    value   : n
  }, {
    address : root + n,
    value : iscore.value(root + n) + 1
  } ];
}
```

If any count becomes greater than two, then the trigger T_E will stop the execution: the score has ended. Else, a new loop iteration is started, and either T_1, T_2 or T_3 will start instantaneously.

Hence we show how a somewhat complex score logic can be implemented with few syntax elements.

Another alternative, instead of putting MIDI data in the score, which makes it entirely automatic and non-interactive, would be to control a screen that displays the part that is to be played. A musician would then interpret the part in real-time, in order to reintroduce an human part to the performance.

6 Conclusion

We presented in this paper the current evolutions of the i-score model and software, which introduces the ability to write interactive and variable loops in a time-line, and the usage of JavaScript to perform arbitrary computations on the state of the local and external data controlled by i-score. This is followed by the presentation of two design patterns for interactive scores, applied to a musical score.

Currently, the JavaScript scripts have to be written in code, even if it is in a generally visual user interface. Given enough testing and user evaluation, it could be possible to have pre-built script presets that could be embedded in the score for the tasks that are the most common when writing a score.

Additionally, we aim to introduce audio and MIDI capabilities in i-score, so that it will be able to work independently of other sequencers. For instance, should it play a sequence of three sounds separated by silence, it would be difficult for the composer if he had to load the songs in an environment such as Ableton Live, and work with them remotely from the other time-line of i-score.

This would also allow for more control on the synchronization of sounds: if they are controlled by network, the latency can cause audio clips that are meant to be synchronized in a sample-accurate manner to be separated by a few milliseconds, it is enough to prevent the usage in some musical contexts.

Acknowledgments

This work is supported by an ANRT CIFRE convention with the company Blue Yeti under funding 1181-2014.

7 References

[1] A. Allombert, G. Assayag, M. Desainte-Catherine, C. Rueda *et al.*, "Concurrent constraints models for interactive scores," in *Proc. Sound and Music Computing 2006*, 2006.

[2] P. Ackermann, "Direct Manipulation of Temporal Structures in a Multimedia Application Framework," in *Proceedings of the Second ACM International Conference on Multimedia*, ser. MULTIMEDIA '94. New York, NY, USA: ACM, 1994, pp. 51–58.

[3] J. Song, G. Ramalingam, R. Miller, and B.-K. Yi, "Interactive authoring of multimedia documents in a constraint-based authoring system," *Multimedia Systems*, vol. 7, no. 5, pp. 424–437, 1999.

[4] M. Toro-Bermúdez, M. Desainte-Catherine *et al.*, "Concurrent constraints conditional-branching timed interactive scores," in *Proc. Sound and Music Computing 2010*. Citeseer, 2010.

[5] N. Hirzalla, B. Falchuk, and A. Karmouch, "A Temporal Model for Interactive Multimedia Scenarios," *IEEE Multimedia*, vol. 2, no. 3, pp. 24–31, 1995.

[6] J.-M. Celerier, P. Baltazar, C. Bossut, N. Vuaille, J.-M. Couturier, and M. Desainte-Catherine, "OSSIA: Towards a unified interface for scoring time and interaction," in *Proceedings of the 2015 TENOR Conference*, Paris, France, May 2015.

[7] P. Baltazar, T. de la Hogue, and M. Desainte-Catherine, "i-score, an Interactive Sequencer for the Intermedia Arts," in *Proceedings of the 2014 Joint ICMC-SMC Conference*, Athens, Greece, 2014, pp. 1826–1829.

[8] C. Böhm and G. Jacopini, "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules," *Commun. ACM*, vol. 9, no. 5, pp. 366–371, May 1966.

[9] H. D. Mills, *Mathematical foundations for structured programming*. The Harlan D. Mills collection, 1972.

[10] K. Kambona, E. G. Boix, and W. De Meuter, "An evaluation of reactive programming and promises for structuring collaborative web applications," in *Proceedings of the 7th Workshop on Dynamic Languages and Applications*. ACM, 2013, p. 3.

Introducing a Context-based Model and Language for Representation, Transformation, Visualization, Analysis and Generation of Music

David M. Hofmann
University of Music Karlsruhe
hofmann@hfm.eu

ABSTRACT

A software system for symbolic music processing is introduced in this paper. It is based on a domain model representing compositions by means of individual musical contexts changing over time. A corresponding computer language is presented which allows the specification and textual persistence of composition models. The system provides an infrastructure to import, transform, visualize and analyze music in respect of individual musical aspects and parameters. The combination of these components provides the basis for an automated composition system capable of generating music according to given statistical target distributions using an evolutionary algorithm.

1. INTRODUCTION

The goal of the presented research project is to yield new findings related to musical composition processes by developing a software system capable of processing and generating music. In order to perform this complex task in a sophisticated manner, further components are necessary: models for music representation, facilities to analyze compositions and an infrastructure to transform results into accurate presentation formats for the user. These modules have proved efficient for symbolic music processing, computer-aided musicology and visualization purposes. This paper demonstrates several applications of the individual components and how they can be combined to tackle the automated composition challenge.

2. MOTIVATION

While music is traditionally notated, read and analyzed in scores, the proposed system provides alternative models to represent music in which individual musical aspects are encoded separately. In the domain of symbolic music processing, musical compositions are often represented as a sequence of notes and rests. While this might be sufficient for a number of use cases, a more complex model is required for extensive musical analysis and for understanding relations between individual aspects. While a single note or sound itself does not have a great significance, its meaning becomes apparent when considering various musical contexts. These include: metric context, rhythm, key, tonal center, harmonic context, harmonic rhythm, scale, pitch, loudness and instrumentation. It is proposed that musical compositions can be represented as a set of the named parameters changing over time.

3. COMPOSITION MODEL

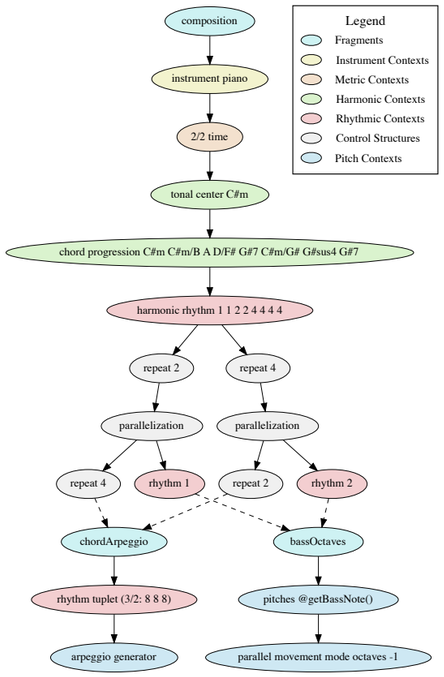


Figure 1. Manually specified context model of the first four measures of Beethoven's *Piano Sonata No. 14, Op. 27 No. 2*, first movement

Models are based on a tree structure containing musical contexts and may further contain so called *modifiers*, *generators* and *control structures*. Modifiers specify how already existing musical contexts are altered in the course of the composition (e.g. transpositions or rhythmic adjustments). Generators are used to create new contexts based on already existing ones (e.g. arpeggios based on chords). Consider Figure 1, which shows a context model of the first four measures of Beethoven's *Piano Sonata No. 14, Op. 27 No. 2*, commonly known as *Moonlight Sonata*.

Each context model has a root node labeled *composition*. Below this node all model elements may be arranged arbitrarily, i.e. it is not prescribed which elements appear on which hierarchy level in the model. In the example, an instrumentation context (*piano*), a metric context (*2/2 time*) and the key (*C#m*) are specified. The model also supports the specification of chord progressions. To supply the durations of each chord, a harmonic rhythm context is required.

Control structures such as repetitions may be nested recursively in order to reflect repeating structures at any level of the composition. For example, in the given model the first repetition branch (*repeat 2*) represents the first two measures. In each measure, the right hand plays four eighth triplets (nested *repeat 4*). Parallel voices are modeled using a control structure named *parallelization*. Otherwise multiple nodes on the same level are interpreted as context sequences. If a subtree contains the same element type on different hierarchy levels, the lower-level context overwrites the higher-level context, allowing to overwrite contexts temporarily.

Models may be split up into so called fragments, which are subtrees that may be referenced from elsewhere. In this way compositions can be specified in a redundancy-free manner since any context combination needs to be specified only once. In the presented example, a fragment named *chordArpeggio* is referenced twice as well as a fragment called *bassOctaves*. The former demonstrates the usage of an *arpeggioGenerator*. It computes a specific chord inversion based on the context harmony and cycles through its pitches in a given sequence. The fragment *bassOctaves* contains a pitch context and a modifier. The pitch context is fed by an expression (prefixed with @) invoking a function called *getBassNote()*, which provides the current bass note of the context harmony. The *parallelMovement* modifier turns single pitches into simultaneously played octaves.

The model captures higher-level concepts which are relevant for compositions such as hierarchical relations and nestings, harmonic progressions on multiple levels and descriptions on how musical material is derived and developed. Instead of enumerating notes, the model is also capable of describing the way music is derived from higher-level building blocks. This aligns with the composition process of human individuals, who generally think in higher-level concepts and structures, many of which this model tries to accommodate. However, a complete list and description of all context types, modifiers, generators and control structures is not possible due to space limitations. Up to now the model was used for representing and processing western tonal music and percussion music, yet it was designed bearing in mind that its application could be extended to atonal music, music from non-western cultures or even electroacoustic music.

Note that there are multiple possible context models for one and the same composition. They can be of explicit nature (comparable to a hierarchical, redundancy-optimized score) or of implicit nature (describing musical development processes). A module for the automatic construction of context models for existing compositions in MIDI or MusicXML format is currently under development. Context models can also be specified and edited manually as explained in the following section.

4. DOMAIN-SPECIFIC COMPOSITION LANGUAGE

```
composition
{
  instrument piano
  {
    time 2/2
    {
      tonalCenter C#m
      {
        chordProgression C#m C#m/B A D/F# G#7 C#m/G# G#sus4 G#7
        {
          harmonicRhythm 1 1 2 2 4 4 4 4
          {
            repeat 2
            {
              parallel
              {
                repeat 4
                {
                  fragmentRef chordArpeggio
                }
                rhythm 1
                {
                  fragmentRef bassOctaves
                }
              }
            }
            repeat 4
            {
              parallel
              {
                repeat 2
                {
                  fragmentRef chordArpeggio
                }
                rhythm 2
                {
                  fragmentRef bassOctaves
                }
              }
            }
          }
        }
      }
    }
  }
}
fragment chordArpeggio
{
  rhythm (3/2 : 8 8 8)
  {
    arpeggioGenerator startInversion 2 startOctave 3
      noteIndexSequence 0 1 2
  }
}
fragment bassOctaves
{
  pitches relative to harmony startOctave 3 findNearestOctave
      true @getBassNote()
  {
    parallelMovement mode octaves -1
  }
}
```

Listing 1. Syntactical representation of the composition model shown in Figure 1

A domain-specific composition language corresponding to the introduced context model allows the textual specification of compositions. This process can also be reversed: algorithmically generated models can be persisted in human-readable text files. Related structured music description languages are *SARAH* [1] and the Hierarchical Music Specification Language [2]. An example is given in Listing 1, which is equivalent to the model in Figure 1. The language infrastructure was built using the framework Xtext [3]. Based on the provided components, an Eclipse-based Integrated Development Environment (IDE) was developed for the composition language featuring syntax highlighting, hyperlinking, folding, outline view and automatic code completion. User interfaces for other features described in the following sections are also integrated.

Stream 1	Instrument (1)	piano											
	Meter (4)	2/2 time											
	Tonal Center (1)	C#m											
	Harmony (8)	C#m											
	Harmonic Rhythm (8)	1											
	Rhythm (48)	12	12	12	12	12	12	12	12	12	12	12	12
	Pitches (48)	G#3	C#	E	G#3	C#	E	G#3	C#	E	G#3	C#	E
Stream 2	Instrument (1)	piano											
	Meter (4)	2/2 time											
	Tonal Center (1)	C#m											
	Harmony (8)	C#m											
	Harmonic Rhythm (8)	1											
	Rhythm (6)	1											
	Pitches (6)	[C#3 C#2]											
	Time (Measures)	1											

Stream 1	Instrument (1)												
	Meter (4)	2/2 time											
	Tonal Center (1)												
	Harmony (8)	C#m/B											
	Harmonic Rhythm (8)	1											
	Rhythm (48)	12	12	12	12	12	12	12	12	12	12	12	12
	Pitches (48)	G#3	C#	E	G#3	C#	E	G#3	C#	E	G#3	C#	E
Stream 2	Instrument (1)												
	Meter (4)	2/2 time											
	Tonal Center (1)												
	Harmony (8)	C#m/B											
	Harmonic Rhythm (8)	1											
	Rhythm (6)	1											
	Pitches (6)	[B2 B1]											
	Time (Measures)	2											

Stream 1	Instrument (1)												
	Meter (4)	2/2 time											
	Tonal Center (1)												
	Harmony (8)	A						D/F#					
	Harmonic Rhythm (8)	2						2					
	Rhythm (48)	12	12	12	12	12	12	12	12	12	12	12	12
	Pitches (48)	A3	C#	E	A3	C#	E	A3	D	F#	A3	D	F#
Stream 2	Instrument (1)												
	Meter (4)	2/2 time											
	Tonal Center (1)												
	Harmony (8)	A						D/F#					
	Harmonic Rhythm (8)	2						2					
	Rhythm (6)	2						2					
	Pitches (6)	[A2 A1]						[F#2 F#1]					
	Time (Measures)	3											

Stream 1	Instrument (1)												
	Meter (4)	2/2 time											
	Tonal Center (1)												
	Harmony (8)	G#7			C#m/G#			G#sus4			G#7		
	Harmonic Rhythm (8)	4			4			4			4		
	Rhythm (48)	12	12	12	12	12	12	12	12	12	12	12	12
	Pitches (48)	G#3	B#3	F#	G#3	C#	E	G#3	C#	D#	F#3	B#3	D#
Stream 2	Instrument (1)												
	Meter (4)	2/2 time											
	Tonal Center (1)												
	Harmony (8)	G#7			C#m/G#			G#sus4			G#7		
	Harmonic Rhythm (8)	4			4			4			4		
	Rhythm (6)	2						2					
	Pitches (6)	[G#2 G#1]						[G#2 G#1]					
	Time (Measures)	4											

Figure 2. Stream model of the first four measures of Beethoven's *Piano Sonata Op. 27 No. 2*, first movement.

5. STREAM MODEL

An alternative representation of musical compositions is provided by so-called *stream models*. These contain parallel timelines for individual musical aspects revealing each musical dimension separately. Stream models provide a novel visual presentation of music suitable for scientific analysis, musicological examination and educational purposes. In Figure 2, a stream model equivalent to the context model in Figure 1 is presented. A context model is transformed into a stream model by means of a compiler. It walks through the tree while resolving references, applying modifiers and generators and finally aggregating contexts at the leaf nodes resulting in a set of separate streams for each musical aspect. Stream models can be considered as expanded context models.

6. SCORE TRANSFORMATION

The infrastructure also provides the functionality to convert stream models into human-readable scores. This is achieved by another compiler, which currently supports LilyPond [4] output resulting in corresponding MIDI and PDF files. An example score is shown in Figure 9. The LilyPond export is configurable, allowing users to render traditional scores and even lead sheets containing chord symbols and fret board diagrams. Another export module for MusicXML [5] output is planned.

7. ANALYSIS FRAMEWORK

An integral part of the system is an analysis framework suitable for extracting statistical data from musical pieces. A related toolkit for computer-aided musicology named *music21* was developed by Cuthbert and Ariza [6]. The proposed system is based on an extensible, module-based architecture. Each module implements specific analysis algorithms for specific musical aspects. The extracted data is written to comma-separated value (CSV) files. Note that different scopes of analyses are possible: corpus, single piece, individual voice in a piece and section-wise analysis. The modules are described in the following sections.

7.1 Duration and Rhythmic Context Analysis

This module analyzes note and rest duration distributions and the duration ratio of notes and rests. Durations can also be analyzed with respect to other musical parameters. For example, Figure 3 illustrates the analysis of note durations

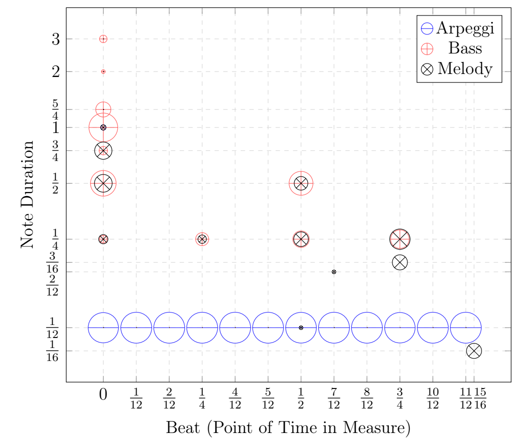


Figure 3. Note duration distribution analysis depending on beats of the complete first movement of Beethoven's *Piano Sonata Op. 27 No. 2*. Circle areas are proportional to the quantity of the data pairs. Triads of the arpeggio notes are equally distributed throughout the measures. The bass and melody voice have selective accentuations, mostly on the first beat of a measure and the following quarter beats. The diagram also reveals that there are certain rhythmic similarities between the melody and the bass voice.

depending on the point of time in the measure they occur for individual voices. With the help of the diagram, rhythmic similarities between voices are visualized. Figure 4 shows an aggregated histogram of note onset times in corresponding measures for various composers based on the analysis of 1093 pieces in MusicXML format. The corpus was compiled by the author from various Internet sources, mainly musescore.org.

7.2 Interval Leap Analysis

This module analyzes the distribution of interval leaps between successive notes in a stream. An example histogram is shown in Figure 7. In case of different note counts (e.g. single notes next to chords or chords with different numbers of notes) the system selects a set of interval leaps with the minimum absolute distance.

7.3 Dissonance Analysis

This module analyzes a piece concerning consonance and dissonance. The dissonance of two simultaneously sounding notes is calculated by taking the frequency proportions of the fundamentals into account. For example, the interval of a fifth is characterized by the frequency ratio 3:2. The so

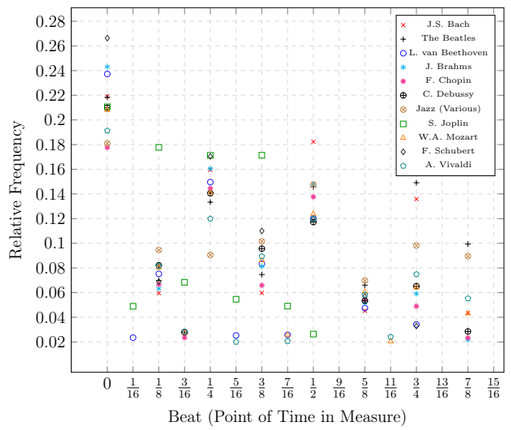


Figure 4. Aggregated beat distribution for various composers. Each analyzed composer places most of the notes on the first beat of a measure. The probability of syncopation increases with more modern music. Values lower than 0.02 were filtered out for better clarity.

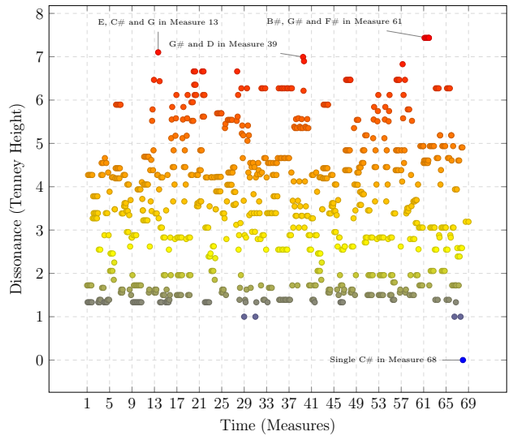


Figure 5. Dissonance analysis of the complete first movement of Beethoven's *Piano Sonata Op. 27 No. 2*

called Tenney Height, which represents a scalar dissonance value for two given tones, is computed using the formula $log_2 ab$, where a and b are nominator and the denominator of the ratio (in this example: $log_2 6 \approx 2.59$) [7]. The dissonance value of a chord consisting of more than two notes is obtained by computing the average dissonance of all note combinations. Figure 5 shows a dissonance plot of the first movement of the *Moonlight Sonata*.

7.4 Harmonic Analysis

The harmonic analysis module focuses on simultaneously sounding notes in order to determine the histograms of keys and harmonies used in the composition. The module also computes a *chord compliance ratio* by dividing the number of notes which are part of the context harmony by the total number of notes. This value is typically high for accompanying voices providing context chords. Moreover the *scale compliance* is analyzed, which is defined as the ratio between notes belonging to the context scale divided by the total number of notes. This is analyzed with respect to the context harmony and the tonal center. Tonally simple and coherent pieces are typically characterized by high scale compliance ratios. The module also supports the analysis and visualization of chord progressions in a directed graph as shown in Figure 6.

8. EVOLUTIONARY COMPOSITION GENERATION

The previous section was concerned with extracting statistical data from existing compositions. For music generation, this process is reversed: statistical distributions are given as input and the system generates compositions that adhere to these requirements as accurately as possible. The space of representable compositions is huge, so a brute-force search would not yield accurate results in an acceptable time. Therefore, so called evolutionary algorithms are used. Programs of this kind have successfully been applied to specific musical problems such as evolving jazz solos [8], rhythms [9], chord harmonization [10, 11] and automated composition systems [12, 13, 14].

8.1 Algorithm Specification

The system creates an initial generation of compositions randomly. Every model is evaluated by compiling it to a stream model (explained in section 5) and analyzing it statistically (see section 7). All distributions are compared with the desired input distributions and all absolute deviations are added up. The goal of the evolutionary process is to minimize the total deviation to zero, effectively implementing a multi-objective optimization [15]. This is achieved by recombining subtrees of compositions selected considering their fitness measure. Mutations are performed by adding, modifying or removing nodes or subtrees. This technique can be considered a special form of *Genetic Programming* [16]. The algorithm benefits from the design of the context model (described in Section 1), as "composing" is now no longer a matter of concatenating notes, but a matter of assembling and restructuring context trees, in which each individual aspect of the music is accessible separately. Results are persisted in text files (the syntax of which was introduced in section 4) and are subsequently transformed to scores (see section 6).

8.2 Algorithm Input and User-defined Constraints

Additional constraints to limit the search space may be supplied optionally in form of an initial context model. It may contain fragments to be incorporated into the composition or predefined constraints such as time signatures, instruments or chord progressions. For this purpose, nodes or subtrees can be marked as *fixed* indicating that they are not to be modified during the evolutionary process.

As an example, it is demonstrated how a blues composition can be generated. The model is shown in figure 8. The predefined nodes specify a basic twelve bar blues pattern as a constraint space. Two parallel voices are defined, one of which is an accompaniment and the other one is the lead voice. The accompaniment is defined in a separate

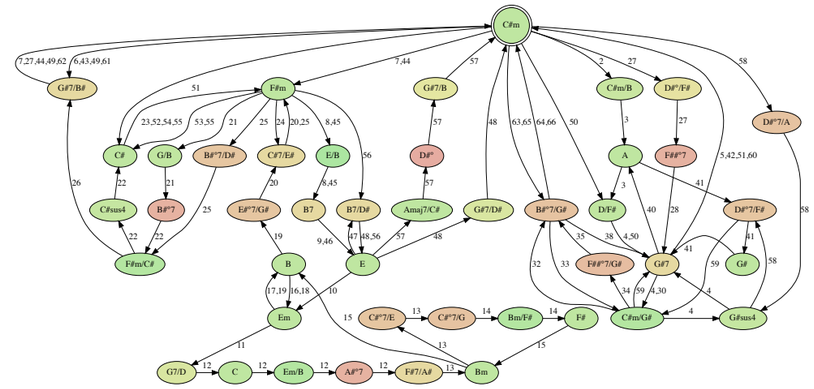


Figure 6. Chord progression graph of the complete first movement of Beethoven's *Piano Sonata Op. 27 No. 2*. The numbers identify the measures in which the respective chord transition was detected. The colors of the nodes correlate with the dissonance value of the chord (green: consonant; red: dissonant).

fragment and consists of alternating fifths and sixths. Literal numbers in pitch contexts are interpreted as degrees on the context scale. Lastly, a final chord (C^7) is predefined. The lead voice is generated by the evolutionary algorithm.

Besides the optional initial context model, the algorithm requires statistical target distributions as input. The following target distributions were set: The desired note duration ratio of the lead voice was set to 95%, meaning that only 5% of the generated material should consist of rests. The requested scale compliance ratio was set to 100%. Furthermore, two target distributions were supplied. The first distribution demands that about 30% of the generated notes should be quarter notes, 50% eighth notes and 20% sixteenth notes. Additionally, the interval leap distribution shown in Figure 7 was given as a target. Of course, more complex combinations of statistical target distributions may be used.

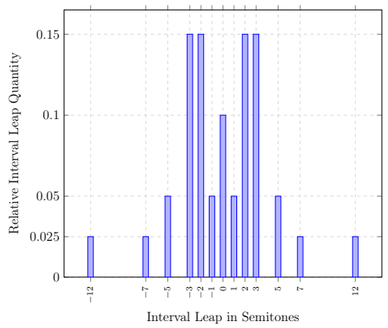


Figure 7. Symmetric Target Distribution for Interval Leaps

9. RESULTS

The computer automatically generated the model shown in Figure 8. The corresponding score is shown in Figure 9. Statistical target distributions and deviations are listed in

Fitness Function	Target	Distance
Note Duration Ratio	0.95	0.005
Scale Compliance	1.0	0
Note Durations	30% quarter notes 50% eighth notes 20% sixteenth notes	0.1
Interval Leaps	see Figure 7	0.27
Total	0	0.375

Table 1. Fitness functions, target values and deviations from the optimum values for the generated blues composition

Table 1. Most listeners found this short piece to be musically pleasant and entertaining. Overall, generated compositions are musically appealing at times. Nonetheless, this is not guaranteed even when using the same set of input parameters repeatedly. The rate of pleasant compositions depends on the number of target distributions and the number of initially predefined constraints. When specifying too few targets, the divergence of both the musical style and the musical quality increases.

Currently the implementation is only capable of optimizing against target features computed for the whole piece resulting in rather monotonous music. The goal is to extend the system in such a way that individual sections of the piece are optimized against a set of section-wise defined target distributions. This could potentially result in a system generating interesting and diverse music.

Another limitation is that the algorithm often does not find an optimal solution, as the search space is very large. Another reason for this might be that mutation and crossover operators still need to be improved. Even though the generated solutions might not satisfy all statistical criteria in all cases, they nonetheless can be appealing and interesting. Eventually, the fact that statistical expectations are not entirely met can contribute to a certain naturalness of a computer-generated composition, causing unexpected musical twists and variety in the music.

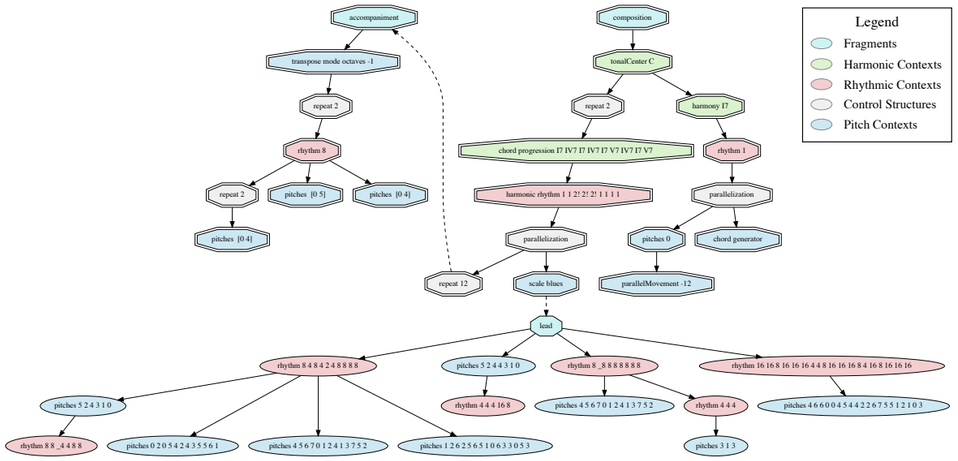


Figure 8. Context model resulting from an evolutionary composition process. Nodes surrounded by octagons were predefined. The subtree below the *lead* node was generated by the evolutionary algorithm.

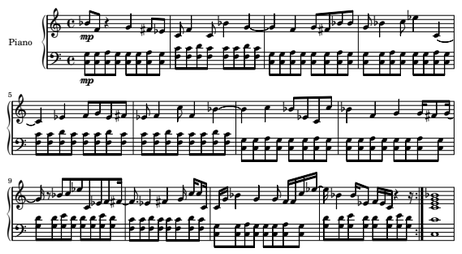


Figure 9. Score of the generated blues composition

10. CONCLUSIONS AND FUTURE WORK

A software system for symbolic music processing was introduced and the functionality of its components was explained for various use cases including context-based music representation, transformation and statistical analysis. Furthermore the combination of these components to form an automated composition system was demonstrated. Future research will be conducted regarding combinations of statistical distributions resulting in appealing musical outcomes. The system will be further improved by providing the functionality to supply different sets of target distributions for multiple sections of a piece. The goal is to develop a higher-level algorithm creating section-wise target distributions and then use the proposed algorithm to generate musical material. Another future goal is to develop a graphical user interface for the composition module allowing users to specify criteria of the desired musical output.

Acknowledgments

This research is generously funded by a scholarship granted by the state of Baden-Württemberg. The author would like to thank his supervisor Prof. Dr. Thomas Troge, the five anonymous reviewers and the paper chair Hans Timmermans for their support and valuable feedback on this paper.

11. REFERENCES

[1] C. Fox, “Genetic hierarchical music structures,” in *Proceedings of the 19th International FLAIRS Conference*. AAAI Press, 2006, pp. 243–247.

[2] L. Polansky, P. Burk, and D. Rosenboom, “Hmsl (hierarchical music specification language): A theoretical overview,” *Perspectives of New Music*, vol. 28, no. 2, pp. 136–178, 1990.

[3] S. Efftinge and M. Völter, “oaw xtext: A framework for textual dsls,” in *Workshop on Modeling Symposium at Eclipse Summit*, vol. 32, 2006, p. 118.

[4] H.-W. Nienhuys and J. Nieuwenhuizen, “Lilypond, a system for automated music engraving,” in *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, vol. 1, 2003.

[5] M. Good, “Musicxml for notation and analysis,” *The virtual score: representation, retrieval, restoration*, vol. 12, pp. 113–124, 2001.

[6] M. S. Cuthbert and C. Ariza, “Music21: A toolkit for computer-aided musicology and symbolic music data,” in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, Utrecht, The Netherlands, August 9-13 2010, pp. 637–642.

[7] M. Deza and E. Deza, *Encyclopedia of Distances*. Springer, 2013.

[8] J. Biles, “Genjam: A genetic algorithm for generating jazz solos,” in *Proceedings of the 1994 International Computer Music Conference*. San Francisco: ICMA, 1994, pp. 131–137.

[9] D. Horowitz, “Generating rhythms with genetic algorithms,” in *Proceedings of the 1994 International Computer Music Conference*. San Francisco, CA: ICMA, 1994, pp. 142–143.

[10] R. McIntyre, “Bach in a box: The evolution of four part baroque harmony using the genetic algorithm,” in *Proceedings of the IEEE Conference on Evolutionary Computation*, vol. 14(3). New York: IEEE Press, 1994, pp. 852–857.

[11] A. Horner and L. Ayers, “Harmonization of musical progressions with genetic algorithms,” in *Proceedings of the 1995 International Computer Music Conference*, San Francisco, 1995, pp. 483–484.

[12] A. Horner and D. Goldberg, “Genetic algorithms and computer-assisted music composition,” San Mateo, CA, pp. 437–441, 1991.

[13] B. Jacob, “Composing with genetic algorithms,” in *Proceedings of the 1995 International Computer Music Conference*. San Francisco, CA: ICMA, 1995, pp. 452–455.

[14] ——, “Algorithmic composition as a model of creativity,” *Organised Sound*, vol. 1(3), pp. 157–165, 1996.

[15] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[16] R. Poli, W. Langdon, N. McPhee, and J. Koza, *A Field Guide to Genetic Programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008.

Sound, Electronics and Music: an evaluation of early embodied education

Lauren Hayes
Arts, Media + Engineering
Arizona State University
Tempe, AZ 85287
laurensarahhayes@gmail.com

ABSTRACT

Discussions of pedagogical approaches to computer music are often rooted within the realm of higher education alone. This paper describes Sound, Electronics and Music, a large-scale project in which tutelage was provided on various topics related to sound and music technology to around nine hundred school children in Scotland in 2014 and 2015. Sixteen schools were involved, including two additional support needs schools. The project engaged several expert musicians and researchers to deliver the different areas of the course. A particular emphasis was placed on providing a form of music education that would engender creative practice that was available to all, regardless of both musical ability and background. The findings and outcomes of the project suggest that we should not be restricting the discussion of how to continue to educate future generations in the practices surrounding computer music to the university level. We may be failing to engage an age group that is growing readily familiar with the skills and vocabulary surrounding new technologies.

1. INTRODUCTION

There is a growing body of literature describing different ways of progressing within higher education (HE) music pedagogy, as many courses in music technology (MT) and electronic music begin to mature. There has been a rapid increase in the numbers of such programmes over the last fifteen years [1]. Undergraduate courses offering instruction in the history and practice of computer and electroacoustic music can be found in universities worldwide. Some of the most recent developments in pedagogy in this area include incorporating research-led teaching perspectives [2], advocating for extra-curricular interdisciplinary collaboration [3], stressing the importance of reflective writing in addition to musical practice [4], along with numerous accounts of existing courses from HE institutions around the world (see, for example, [5, 6]).

The large-scale project *Sound, Electronics and Music* developed out of the author's recent observations of undergraduate MT courses: by the time students undertake introductory modules in digital sound within universities, many of them are already familiar with, if not highly practiced

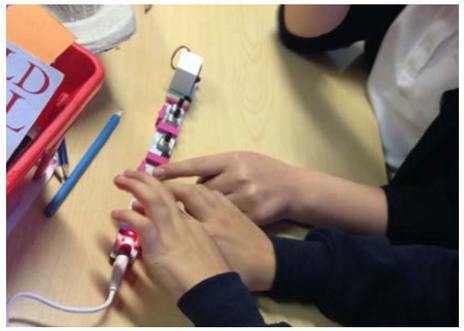


Figure 1. A collaborative performance on a KORG littleBits Synth Kit.

in working with digital audio workstations and electronic sound production techniques. For example, within the undergraduate student cohort of the BA in Digital Culture offered by the School of Arts, Media and Engineering at Arizona State University, many first year students commence the course already in possession of such skills [1]. Several students are already producing their own electronic music by working with digital audio software, such as FL Studio. A symptom of this is that they often harbour the aesthetic determinism that commercial software can foster. Equipped with open-access, affordable software and an internet connection "millennials delve into an individualized creative process with their preferred tools at arms reach" [7].

On the other hand, computer science and engineering are being marketed to younger children through low-cost computer hardware such as the Raspberry Pi [2], and electronic inventor kits including, for example, littleBits [3]. With the advent of touch-screen technology within mobile phones and tablets, many children are becoming technically engaged at a very young age: viral videos circulate on social media sites of one-year-olds using hand gestures observed from parents to interact with touch-screens. Even technology within schools has become ubiquitous, and standardised to some extent. All of the schools involved in this project are government funded, and use the same laptops and smartboard projectors on a daily basis in every classroom for curricular teaching. Computers are not only used by staff, but are often distributed among pupils. Schools

[1] Class survey taken at the start of the course MDC211 Introduction to Digital Sound in 2015 and 2016.
[2] https://www.raspberrypi.org/
[3] http://littlebits.cc/

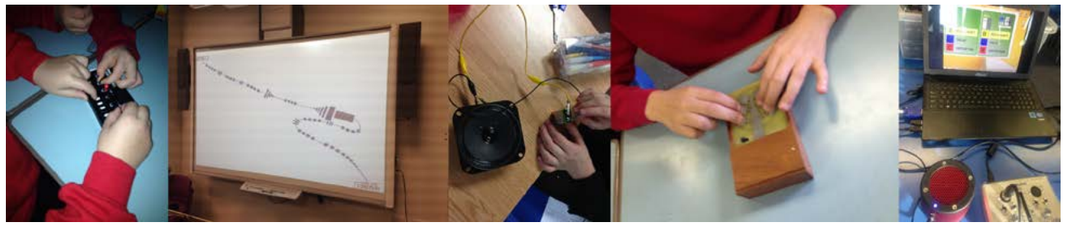


Figure 2. Pupils performing with graphic scores and hardware-hacked, commercial, and software instruments.

continue to foster traditional music education, which encompasses theory, aural skills, musical notation literacy, and instrumental training, yet there is clearly a technocultural space in which to develop a pedagogical approach to MT. Acknowledging this potential curricular opportunity—along with current leanings towards STEM education [8]—was key to the development of this project.

Sound, Electronics and Music was conceived as a ten-week programme. The aim of the project was to harness this new potential for accessible music education, which could engage pupils regardless of their musical and socioeconomic backgrounds. The project was funded for two consecutive years by Creative Scotland's Youth Music Initiative, which is aimed at providing high-quality musical activities for young people in Scotland. The programme was offered to around nine hundred 8-12 year old children in sixteen schools in West Lothian, Scotland. Sixteen one-hour weekly workshops were given in eight schools each week (two classes per school, eight schools per year). The course was offered to Primary 5-7 classes in the first year, and was expanded to include two after-school mixed secondary classes in the second year. Two additional support needs (ASN) schools were involved in the project and received all of the same course material as the other schools.

The curriculum was designed by the author, who was joined each week by a different musical practitioner. Each guest was given the freedom to contribute a unique perspective and set of skills. Of the seven additional musicians involved, five have completed or are currently completing doctoral studies in sound and music-related topics. In addition to the large number of workshops given, the project produced four new software applications designed in Max/MSP, which were distributed and remain on laptops within the schools.

2. OBJECTIVES

2.1 Inclusive Classrooms

The course was devised to inspire creative exploration from all pupils, particularly those who had no formal training in playing a musical instrument or reading traditional musical notation. Working with sound as a material—and using materials to make sounds—provides a non-preferential platform from which to create music. The experience of sound itself—how it is perceived, understood, and talked about—can be considered without necessarily having to engage with the solfège system, rhythm analysis, and so on. However, pupils who *were* receiving music lessons were encouraged to bring their instruments to the classes so that they could employ and expand these skills. Working from an experimental perspective, they were introduced to extended techniques, improvisation, and electronic augmentation.

Due to the exploratory approach taken in the workshops, there was very little modification needed to include pupils at the ASN schools. These sessions covered the same material, but were flexible in their delivery, allowing more time for exploratory play. The use of narrative was a helpful device here as it could be used to thematise the weekly sessions.

2.2 Accessibility and Legacy

The project acknowledged that, while young people's affinity with technology is often purported to be fact, this is not a universal phenomenon within the UK, and can be directly related to socioeconomic status [9]. Lack of formal musical training or musical literacy among children can often be linked to low family income [8]. As such, the course was designed to work with technology that would always be available in class (school laptops and a smartboard projector), as well as utilising low-cost hardware and found materials. It was important to ensure that what was taught could be developed further outside of the classroom. Every new instrument or piece of equipment that was introduced was either available to purchase online at a low cost, or could be found in local hardware stores. This turned out to be crucial to the legacy of the project because children would often ask where they could acquire materials after each session. Despite the young ages involved, pupils would often enquire about audio programming languages, particularly after they had used software that had been designed specifically for the course. They were directed to open-source software (OSS) such as Pure Data and ChucK.

Each school was provided with a box of sound equipment. This contained a variety of items that were showcased during the weekly workshops. The kit included:

- a two-channel soundcard
- headphones
- a KORG littleBits Synth Kit
- two Makey-Makey invention kits
- a Minirig loudspeaker
- a Zoom H1 portable sound recorder
- microphones and stands
- cables
- DIY synthesizers, electrical components, batteries, and speaker cones.

A manual was left in each class, outlining ideas for lesson

Figure 3. Open experimentation with the Victorian Synthesizer.

plans, as well as providing detailed descriptions of how to connect and operate all hardware, and instructions for running the provided software.

2.3 Supporting Teachers

Recent research into the role of technology within education stresses the importance of getting teachers involved in the learning process: "Educators have to be willing to learn about and engage with new technologies so that, as with any discipline area, they are aware of new developments and how these can be used to inform the learning environment" [9]. In order to ensure that lesson content could be repeated and expanded upon, it was important to involve teachers from the outset. Continuing professional development (CPD) training was provided outside of the scheduled class time.

Many of the concepts involved in the course were new to both the class teachers, as well as the music teachers that were present. Out of the sixteen schools visited, two provided a music teacher, instead of the class teacher, to supervise the class. Out of the thirty-two classes involved, only one teacher had previously worked with any of the technology that was being used (the Makey-Makey invention kit). Another teacher was in the process of developing a new MT course for his secondary school pupils.

Teachers were offered two CPD sessions: one at the start, and one at the end of the course. These were an opportunity for teachers to spend more time familiarising themselves with the software and audio equipment. It also allowed them to discuss ways in which they could continue to foster the various skills developed during the workshops.

3. EMBODIED LEARNING

The broad range of relevant topics that could be taught at the school level has been documented elsewhere [10]. Rather than prescribing a particular set of lesson plans, this section expounds upon some of the key themes that were prevalent throughout the conception and execution of this project.

3.1 Course Design

The course comprised ten workshops. Learning was scaffolded by building upon the previous weeks' learned skills and vocabularies. In this way, a sense of continuity was established from week to week. Furthermore, at the start of each class, pupils were encouraged to present examples of sounds they had heard outside of the class via descriptions or recordings. These sounds were used both as material for listening exercises, and as samples for sound organisation and manipulation. Pupils were able to directly contribute their own material to the course.

The majority of the workshops were designed to facilitate embodied learning where possible. This draws on current research into embodied cognition, which is rooted within the philosophy of Maurice Merleau-Ponty. Merleau-Ponty suggests that it is through perception that we engage with the world, but that perception is linked to action itself, being something that we *do* [11]. Research into skill-acquisition [12] and, more recently, practice-based learning theory also stresses the important of the role of the body: "To the extent that learning/ knowing is a matter of doing, doing can only be performed through the efforts of the human body" [13].

3.2 The Practice of Listening

Listening was fostered as a core skill throughout the sessions. Pupils were encouraged to develop their listening practice both in and out of school. Working with Sound and Music's *Minute of Listening* [4] software, which is commercially available and has been specifically designed to be used in classrooms, pupils were given a space in which to focus on their perception of sound. They were asked to describe the sounds they heard, whether natural or synthetic, and were urged to develop vocabularies to describe these sounds. Opposite word pairings, such as loud and quiet, rough and smooth, were offered as prompts. Pupils quickly identified that many sounds lie on a continuum: for example, a recording of cricket chirps is actually made up of numerous short sounds.

In the vein of Pauline Oliveros' Deep Listening practice, pupils were encouraged to listen to sounds from daily life and nature, as well as silence. Interestingly, several pupils claimed that they recognised many of the abstract sounds played to them during the listening exercises. Computer games and film soundtracks were cited as the source of this familiarity. As Oliveros points out, developing a listening practice contributes to creativity and communication skills: "It cultivates a heightened awareness of the sonic environment, both external and internal, and promotes experimentation, improvisation, collaboration, playfulness and other creative skills vital to personal and community growth. Plus it's a ton of fun" [14]. Listening exercises also required that the pupils developed an awareness of their bodies. They were asked to consider their posture, how much they were fidgeting, how still they could sit while listening, and whether particular sounds made them feel relaxed or agitated. They were also asked to experiment with both eyes-open and eyes-closed listening.

3.3 Authoring Sounds

Having developed an awareness of listening as a practice, pupils were given portable sound recorders. Tasked with collecting different sounds from around the school and grounds,

[4] http://www.minuteoflistening.org/

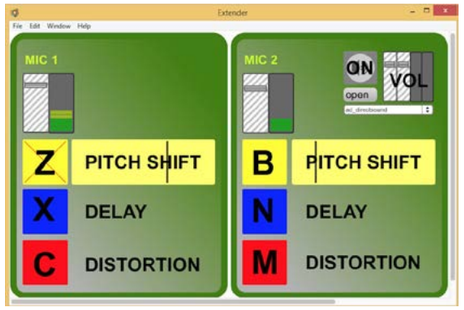


Figure 4. GUI of Max/MSP app for augmenting acoustic sounds-makers.

the pupils were given free reign to experiment. They were shown how to excite different objects and materials, and how to work with the combination of headphones and a sound recorder to zoom in on sounds that may not have been deemed interesting without focused listening. This form of embodied learning enabled pupils to move around the school, seek out new sounds, discover interesting action-sound combinations, and take on a truly investigative role. This supports Mark Johnson's claims about the importance of artistic investigation: "the value of an artwork lies in the ways it shows the meaning of experience and imaginatively explores how the world is and might be primarily in a qualitative fashion. Therefore, art can be just as much a form of inquiry as is mathematics or the empirical sciences." [15].

The Zoom H1 recorder was used because nearly all recording can be done using a single start/stop button. After the sound collecting was completed, pupils would play back the recorded samples to each other in order to guess and describe the sounds that had been gathered. They then would discuss how these sounds could be transformed into music. The collected sounds were reviewed, categorised, and named. Working with the smartboard projector, pupils were encouraged to use their recorded sounds as compositional material within several specially designed Max patches. One of these was devised to allow a collaborative class composition. Pupils would collectively vote on several variable sound parameters. These included selecting a part of the sample to be played back and looped, changing the pitch, or adding an amplitude envelope over the duration of the looped sound. The pupils would quickly determine which settings would produce the most interesting, or indeed most humorous results. For example, speeding up the sound of recorded speech, particularly when it was that of the teacher, was often requested.

These sample libraries grew throughout the sessions as pupils contributed their own recordings from out with the workshop time. Each class collectively defined their own unique aesthetic.

3.4 Making and Hacking

As Nicolas Collins points out in his book on hardware hacking, computers can be an awkward interface and "sometimes it's nice to reach out and touch a sound" [16]. Working with classes of between twenty and thirty primary school children would not suit a model where each pupil was working individually on a computer. Furthermore, time spent focusing on the smartboard projector had to be limited in order to keep attention. Collins' philosophy seemed even more fitting in this context: "The focus is on soundmaking performable instruments, aids to recording, and unusual noisemakers... the aim is to get you making sounds as soon as possible" [16].

By making new instruments and hacking existing devices, pupils were encouraged to use their imagination and discover new affordances of objects. Junk materials such as paper tubes, water bottles, and elastic bands were turned into acoustic sound makers. Makey-Makeys were connected to fruit, conductive tape, pencil graphite, and chains of the pupils' own bodies as a means to trigger sounds. Pupils devised modifications to John Bowers' *Victorian Synthesizer* [16] by sending electrical signals through sharpeners, spectacles, and their classroom furniture (see Figure 3). This process of appropriation enabled the pupils to gain authorship of their new instruments, and also become an intrinsic part of their deployment. This type of playful embodied learning, involving the manipulation of physical objects, has been proven to enhance learning [17].

3.5 Improvisation and Collaboration

Improvisation was used within many of the sessions as a way to help the pupils make sense of the wide array of new sounds that were being produced. On the occasions that they were not forthcoming with their music, pupils were encouraged to play works such as John Steven's *Click Piece* [18], or create and then perform graphic scores for each other (see Figure 2 for some of the graphic scores and instruments used for improvisation). By being non-prescriptive about the aesthetic outcomes, there emerged a "space for open-ended inquiry, an investigation of cause and sounding effect" [8].

Collaborative working was encouraged. This took the form of whole class collaboration, where decisions on how to sculpt a piece, or select samples to use, were made either through voting, group discussion, or turn taking. Small group collaborations also enabled instrumentalists to work with newly-appointed *live electronic performers* who would manipulate sounds made by their classmates through a Max patch that could be operated swiftly using a computer keyboard and trackpad (see Figure 4). Acoustic instruments, voice, and found-material sound makers all were pitch-shifted, distorted, and delayed. Further collaboration took place within the physical instruments themselves, where often two or more players would perform on a single instrument at once. For example, when playing the KORG littleBits, one performer would select pitches, while another would open and close the filter (see Figure 1). Other forms of collaboration were established by the pupils themselves: performances would often feature clapping, singing, speech, conduction, or in the case of the Makey-Makey sessions, movement and whole body contact.

4. EVALUATION

> *I loved it because its two of my favourite things, tech and music, together.* [P]

The delivery of the project was evaluated by post-workshop surveys, which were distributed to all class teachers. Teach-

ers were invited to assess various aspects of the course such as the professional delivery of the sessions, as well as its impact on the education, skill acquisition, and health and well-being of the pupils involved. This was done through a rating system. This was combined with qualitative evaluation, which took the form of written comments from both pupils [P] and teachers [T] on the feedback forms. Additionally, teachers were invited to further expound on their opinion of what had taken place in the final CPD sessions.

One of the most common themes that appeared within the feedback—from both teachers and pupils—was the value of the interdisciplinary nature of the course:

> *This has been an excellent series of workshops, delivered in a interesting and interactive way. The pupils have all responded very well to them, exposing them to a wide range of skills and experiences (not limited to music - but includes some science etc).* [T]

Teachers also noted that while much of the material was also new to them, they were confident that many of the skills learned could be applied to other subject areas:

> *As a teacher I have learned a lot as it was not an area I knew much about. I now feel I have new knowledge and skills that I can use with future classes and the workshop has demonstrated good links between different areas of the curriculum (music and science).* [T]

In addition to identifying potential links with other academic areas, teachers commented on the benefit to the social and communication skills of the pupils:

> *[S]ome of the sessions delivered were cross - curricular. E.g. Science with electricity, Health and Well Being and how music can make you feel different emotions, Writing and responding, talking and listening amongst others.* [T]

The course was also successful in the two secondary schools in which it was delivered. Teachers remarked on how it complimented new MT courses that were being introduced:

> *I have already informed the Music departments in all WL secondary schools about the experience and have recommended it... The subject matter was a departure from the normal curriculum delivered in the Secondary Music curriculum and this complimented the Music Technology course that we have introduced this year at Nationals level.* [T]

The scope for experimentation and the hands-on approach, they suggested, could support the more individualised computer-based work that had recently been implemented in the curriculum.

Many teachers remarked on how the workshops seemed to appeal to those children who would not usually engage in group work, as well as those who often struggled in class:

> *Over the weeks I have witnessed some pupils being able to demonstrate their abilities in this area who find engaging in some academic work challenging.* [T]

Involving a range of different practitioners to deliver the workshops gave the pupils a broad view of existing practices within experimental and computer music. Unsurprisingly, the pupils were most responsive to the more hands-on workshops such as hardware hacking:

> *The pupils all really enjoyed the workshops and were enthusiastic to learn new and different ways of making music. They also looked forward to the different special guests who were invited each week to share the expertise in different areas. A really worthwhile project.* [T]

In rating the course, all teachers either agreed or strongly agreed that it had provided their pupils with new transferrable skills, as well as developing their social, emotional, and linguistic capacities. All of the responses to the question about increased employability were either neutral, or deemed the question not applicable. Only a single response addressed this topic:

> *I anticipate pupils being more able to work independently in the expressive arts.* [T]

The younger students gave appraisal by making thank you cards with drawings of their favourite activity. Older pupils gave succinct statements such as:

> *I think this was really fun and I enjoyed it very much #WouldRecommend.* [P]

In addition to the unanimously positive response from the pupils and staff, a further outcome worth noting was that in at least two of the schools, children took the initiative to set up their own electronic sound and music sessions. These took the form of lunch-time clubs where dedicated pupils took ownership of the equipment and would distribute it among other interested parties during the lunch hour. This often resulted in more sound recordings, short performances which were included in the next official workshops, and also further questions about how the equipment could be used.

5. CONCLUSION AND FUTURE WORK

This paper has described the development, implementation, and evaluation of a large-scale pedagogical framework for computer-based and electronic music undertaken within primary, secondary, and ASN schools. This research provides evidence to support the assertion that computer music and MT have a place within the pre-university classroom. This is firstly demonstrated by the overwhelmingly positive feedback and evaluations that were received.

Secondly, experimental musical practice provides an excellent forum for inclusive and embodied learning to take place. By engaging in practices such as listening, sound collecting, recording, hardware hacking, and instrument building, pupils became physically invested in their own learning. As Adam Tinkle suggests, "Rather than relying so exclusively on externally imposed norms and traditions to determine and delimit each step up a childs ladder to musicianship, what if instead music education was self-education in which students were, like citizen-scientists, set loose to probe and document the sounding world?" [8].

This was also supported by the ease with which the course could be implemented within the ASN schools.

Thirdly, this project builds upon related research—where teenagers were given the opportunity to design their own instruments, supported by mentors—that suggests that a participatory approach to music technology can help to generate interest in the broader fields of science and technology across genders [19]. The interdisciplinary applications of the project was evidenced through the feedback received. Nevertheless, recent studies of MT in HE institutions suggest that, despite technology's potential for democratisation, "existing ideologies of gender and technology, and social class differences, are being reinforced or even amplified through music in HE" [1]. Certainly, we must proceed with "careful reflection" [1], while we design MT courses for future generations. One of the teachers involved in *Sound, Electronics and Music* described it as:

> *A fantastic and motivating course... ideal for a very boy-heavy group,* [T]

which clearly suggests that there is still work to be done.

While legacy was an important consideration, further developments could improve this. Developing cross-platform apps that could be shared by teachers on *any* laptop, and using OSS, such as Pd, throughout would also be helpful to maintain continuity. All the sounds and music produced within the course were documented and stored on each class' laptop, with a view to being hosted on their school's website at the end of the course. Due to security restrictions this has not yet been implemented. This would provide further opportunities for pupils to discuss and comment on their peers' work.

Acknowledgments

This work would not be possible without the insight and expertise of Nancy Douglas, who facilitated the entire project through West Lothian Community Arts and astutely identified its potential impact. I would like to thank all of the experts involved for their enthusiasm and imagination: Jessica Aslan, Emma Lloyd, Christos Michalakos, Zack Moir, Yann Seznec, Greg Sinclair, and Shiori Usui. This project was generously supported by Creative Scotland.

6. REFERENCES

[1] G. Born and K. Devine, "Music Technology, Gender, and Class: Digitization, Educational and Social Change in Britain," *Twentieth-Century Music*, vol. 12, no. 02, pp. 135–172, 2015.

[2] J. R. Ferguson, "Perspectives on Research-led Teaching," in *Creative teaching for creative learning in higher music education*, L. Haddon and P. Burnard, Eds. Ashgate Publishing Company, 2016.

[3] E. Dobson, "Permission to Play: Fostering Enterprise Creativities in Music Technology through Extracurricular Interdisciplinary Collaboration," *Activating Diverse Musical Creativities: Teaching and Learning in Higher Music Education*, p. 75, 2015.

[4] D. Moore, "Supporting students in music technology higher education to learn computer programming," *Journal of Music, Technology & Education*, vol. 7, no. 1, pp. 75–92, 2014.

[5] H. Timmermans, J. IJzermans, R. Machielse, and G. van Wolferen, *Education On Music And Technology, A Program For A Professional Education*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2010.

[6] C. Boehm, "Between Technology and Creativity, Challenges and Opportunities for Music Technology in Higher Education (long version/CIRCUS 2001)," *CIRCUS*, pp. 55–72, 2001.

[7] D. Walzer, "Sound Exchange: Reframing Music Composition Educational Practice," *Leonardo Music Journal*, vol. 25, pp. 34–36, 2015.

[8] A. Tinkle, "Experimental Music with Young Novices: Politics and Pedagogy," *Leonardo Music Journal*, vol. 25, pp. 30–33, 2015.

[9] M. Baguley, D. L. Pullen, and M. Short, "Multiliteracies and the new world order," *Multiliteracies and technology enhanced education: Social practice and the global classroom*, pp. 1–18, 2009.

[10] A. R. Brown, *Computers in music education: Amplifying musicality*. Routledge, 2007.

[11] M. Merleau-Ponty, *Phenomenology of Perception (C. Smith, Trans.)*. Routledge and Kegen Paul, 1962.

[12] H. Dreyfus and S. Dreyfus, "The Challenge of Merleau-Ponty's Phenomenology of Embodiment for Cognitive Science." in *Perspectives on Embodiment: The Intersections of Nature and Culture*, G. Weiss and H. F. Haber, Eds. New York: Routledge, 1999.

[13] A. Yakhlef, "The corporeality of practice-based learning," *Organization Studies*, vol. 31, no. 4, pp. 409–430, 2010.

[14] P. Oliveros. Across Boundaries, Across Abilities. [Online]. Available: http://deeplistening.org/site/content/about

[15] M. Johnson, "Embodied knowing through art," *The Routledge companion to research in the arts*, pp. 141–151, 2011.

[16] N. Collins, *Handmade Electronic Music: The Art of Hardware Hacking*. London: Routledge, 2009.

[17] A. S. Lillard, "Playful learning and Montessori education," *American journal of play*, vol. 5, no. 2, p. 157, 2013.

[18] J. Stevens, J. Doyle, and O. Crooke, *Search and reflect: A music workshop handbook*. Rockschool, 2007.

[19] A. Thaler and I. Zorn, "Music as a vehicle to encourage girls' and boys' interest in technology," in *5th European symposium on gender & ICT. Digital cultures: participation–empowerment–diversity*, 2009.

Performing Computer Network Music.

Well-known challenges and new possibilities.

Miriam Akkermann
Bayreuth University
miriam.akkermann@uni-bayreuth.de

ABSTRACT

In this paper, the focus is set on discussing performance issues of projects using computer networks. Starting with a rough overview on most mentioned projects and classifications from the 2000's, well-known technical challenges are gathered, providing the base for reflections on new options for performing Computer Network Music.

1. INTRODUCTION

Using computers as well as using computer networks in music production is nothing uncommon. New interfaces for the use of wireless technologies have been established and connection a controller via wifi is nothing difficult. There also have been many reflections on computers as instruments, tools, or the computer within musical productions in general. Computer Network Music, however, still seems to be a niche category. This impression may come from the fact that the average use of technology overtook the genre before new role models for Computer Network Music were established. What was the significant for this category? Are there recent developments? What could performing Computer Network Music in the future mean?

2. COMPUTER NETWORK MUSIC

The term 'computer network music' was defined by Scot Gresham-Lancaster as the "enclave of experimental composer/performers who have worked consistently to use the latest breakthroughs in musical hardware and software advances."[1] This subsumes a wide range of very diverse projects with various technical and artistic directions. The term includes 'network', implicitly 'data network', which can stand for the (physical) network system that connects data stations, the structure of a data network, or entitle the entire setup.[2] The particular definition of the term depends on the special field, but also on the single example or established system.

The use of these networks can follow different concepts, e.g. connecting computers during the performance, using the network to share or distribute music, enabling collaborative work on music, as well as hybrid forms of these concepts or combinations with other systems.

Golo Föllmer defined 'Netzmusik' – 'music in the internet' – as music that reflects the specific characteristics of the 'net'. 'Internet' for him had become an undefined entity constituted by individual connected computers.[3] Also Peter Manning talked about 'internet music' or 'internet-based music networks', referring to 'internet' defined as the world wide web. He therefore did not consider projects using other network systems.[4]

In consequence, the term Computer Network Music can be used for an inhomogeneous group of music works or musical performance which have in common that using computer networks is a basic requirement. The diversity of concepts are also reflected in the mentioned historic references.

3. HISTORIC REFERENCES

A significant high number of the papers on computer network music were published by artists or technician involved in this field. The authors used historic references mainly to outline classification systems as well as outlining the context of their own work. Depending on the context, they referred to artists, compositions, network systems and ensembles.

One of the most mentioned reference was John Cage. Gil Weinberg stated in his article, that in *Imaginary Landscape No. 4* the first interdependent musical network was created. It connected musicians via a net on radio stations, and two musicians at a time wit one single radio. His second example was *Cartridge Music*, where, according to Weinberg, "Cage made his first attempt at a musical network focused on tactile generation of sounds and intra-player, amplification- based interdependencies."[5]

Another important figure was Max Neuhaus. Peter Manning especially emphasized the cutting-edge role of Neuhaus' radio works showed end of the 1960's and beginning of the 1970's.[4]

Also William Duckworth mentioned Cage and Neuhaus, but he took *Imaginary Landscape No 5*, *Williams Mix,* and *Sounds of Venice* as reference for interactive music, and Neuhaus for projects related to cell phones and satellites. Additionally, he quoted the ensembles *The League of Automatic Music Composers* and *The Hub* in his chapter „Music on the Web". [6] These ensembles can also be found in Manning's book. Here, however, they were presented as pioneers of „Laptop Music and Related Activities", not for music in the internet.[4]

Föllmer bridged these differences in proposing historic lines categorized by the artistic aim, such as academic line, line of media art, performance, pop and mixed media art. He classified for example *The League of Automatic Music Composers* and *The Hub* as part of the academic line, which usually used the term Computer Network Music. These non-profit projects were usually hosted by universities or research centers, and did not necessarily use similar technologies or structures. Following Föllmer, most of these projects share the reference to Cage, as this provided the link to the already established fields Algorithmic Composition and computer music.[3]

4. WELL-KNOWN CHALLENGES

The diversity within this field cannot only be seen in the various categorizations but was also reflected in Föllmers overview on existing projects in 2005. In the *Computer Music Review* Issue 6 from the same year, "Internet Music" was picked out as the central theme.[7] Hugill introduced the the topic in claiming that

> "[t]his issue sets out to be a primary source for all aspects of this subject. Internet music is, almost by definition, international and interdisciplinary, so the authors from several countries include: composers and musicians, computer scientists and cultural theorists, experts in intellectual property and ideas, multimedia artists and educationalists. The range of topics reflects the diversity and complexity of the Internet itself, and the contents cover the entire spectrum from the highly technical to the highly descriptive."[7, p. 435]

The inhomogeneity of the field became the significant. But even though Hugill sets the focus on music using "internet", the common ground for all projects in this scattered field is the use of computer networks. Basic challenges therefore derive from technical aspects. Over time, there appeared very different strategies of how to deal with those, or embed them into the artistic concept.

4.1 Data transfer rates

From the very beginning, the transferable data volume was a central challenge in computer networks.

The League of Automatic Music Composers

In 1978, John Bischoff, Tim Perkis and Jim Horton founded the Oakland based ensemble ‚The League of Automatic Music Composers'. This ensemble members used directly connected KIM-1 Computer to send and receive small data packages, text messages, which allowed to influence the music systems of the other player. The musicians played all in one room, being able to communicate with the other ensemble members and experiencing directly the sounds produced by the other computers.

The HUB

Still using local networks and simple text data, Bischoff and Perkis founded the ensemble The HUB with Chris Brown, Scott Gresham-Lancaster, Phil Stone and Mark Trayle in 1988. In 1990, Gresham-Lancaster noted the start of The HUB2, as the ensemble was now using MIDI-data instead of text messages. This also allowed the use of MIDI-controller, e.g. MIDI-keyboards as input and MIDI-instruments as output unit in addition to the common computer keyboard. By Pitch-to-MIDI-tracking, even acoustic instruments could be implemented. The data flow as well as the used hard- and software was determined for every piece.

End of the 1980's first experiments took place to use networks connecting musicians which were situated in two different rooms. In 1987, The HUB played a concert with two groups of three people playing at two different locations, one at Clocktower and one at the Experimental Intermedia Foundation. This derived from a practical reason: The concert was founded by these two institutions. The player's computer were connected via modem, the audience was invited to change between the two locations within long pauses. This concert, which was assumed as the first with music via modem by Nicolas Collins, was the starting point for The HUB to work on settings with distant located and via network connected musicians.[8]

In 1997, three years after the foundation of the World Wide Web Consortium in 1994, the ensemble members played a concert in which they altogether controlled the sound synthesis software Grainwave, each being at some place in California. Technically, the set-up worked, but as only control data was exchanged, it was impossible to listen to the produced sounds of the other members.

After having had several attempts on playing concerts using the internet, the ensemble officially split in 1998, but still playing revival concerts in small local networks.

With new network technologies and enhanced network distances, it was possible to connect musicians which acted on far distant locations. Starting in the 2000's, it was also possible to stream audio data via computer networks. Here, the challenge was not only the data volume, but also the transfer speed.

4.2 Dealing with Latency

The time span which was necessary to transfer data between two computer became relevant when processes were intended to happen in real-time. In the first decade of the 21st century, many different case studies and (artistic) approaches and have been established in order to deal with this challenge.

ping

In 2001, Chris Chafe presented *ping*, a project in which data packages were sent in real-time between two server. The concept was to make audible the usually silent server request (ping) and the answer (pong) which confirmed the connection. Following Alexander Carôt, this more installative performance was the first known project that benefited from the practicability to use real-time audio

connections within Internet2.[9] Entitled *SoundWIRE* by Chafe, short for 'Sound Waves on the Internet from Real-time Echoes', the technology used for *ping* also gave name for a research group at CCRMA, in which several tools for bi-directional transmission of uncompressed audio data in real-time were developed. [10]

quintet.net

Another way of dealing with latency established Georg Hajdu in his system for network ensemble. In quintet.net, up to five musicians, a conductor further optional devices are connected to a central server, each computer running the network system software. According to his statement, he was inspired by The HUB when he started to develop his system in 1999.

The system is designed for both, local networks and performances using internet. A conductor sends notation or playing instructions via server to the clients/musicians who then have to follow these instructions and trigger a sound source. Depending on the size and the structure of the network, the latency could influence three data transfers: the time between sending a notation by the conductor until it becomes visible at the clients' screen, from the clients' action and the trigger of the sound source and the duration until the triggered sound can be heard by the audience. Hajdu therefor used a notation system which was based upon John Cages concept of time brackets, aiming at a real-time compositions and improvisation system. The notation implicitly included a start-interval and a relative time code, which also influenced by the latency. The musicians then should perform the notation at a suitable moment, the lack of exact sound control was accepted.[11]

4.3 Technical restrictions and performance issues

Dealing with data transfer and latency also had a big influence on the performances or computer network music. For Carôt, quintet.net was an example for an interactive network performance system with Latency Accepting Approach LAA. This meant, latency was accepted and implemented within the work. The optional deficit created by the technical system was balanced by the structural arrangement of the performance system or single work. The counterpart to this concept was for Carôt the Realistic Jam Approach RJA. Here, interactions in real-time between the involved musicians are central, therefore was high transfer speed and best network quality fundamental. The goal was to create systems which established a situation as close as possible to a live situation. Carôt himself developed *Soundjack*, a RJA software application which was inspired by *SoundWIRE*. It allowed to stream audio directly peer-to-peer.[9]

As RJA systems simulated live situations, the main domain was real-time use in the internet. They were hardly used for public concerts, but promoted as spaces for online jam sessions.

In concert context, LAA systems seem to be more present. The European Bridge Ensemble for example, which is specialized on compositions for quintet.net, plays concerts on a regular base. Also the ensemble The HUB could be categorized as latency accepting. Föllmer took this ensemble as the example for a performance oriented project.[3] However, The HUB did focus on performances in local networks, which does not produce as long latency as e.g. long distance internet connections.

5. NEW OPTIONS

With the recent network standards, latency can be reduced in physically wired local networks to almost zero. The possible data transfer rate has increased, so that real-time processes are no longer limited to control data but can also provide real-time audio transfer.

Not technically new, but also not yet discussed in the field of computer network music is the use of Wifi devices and tablet computers or smartphones. This may derive from the categorizations already outlined under historic references. As said, Duckworth mentioned Neuhaus' radio works related to the projects subsumed under 'cell phones and satellites', whereas Föllmer did not mention Neuhaus but categorized radio projects as media art. Mobile devices seem to be assigned to radio works and – in consequence – more likely discussed in media art and not music context.

5.1 New Combinations and Interaction

When using the initial definition of computer network music, this includes all works using connected computer units, which technically includes also mobile devices. This opens the view on a wide field of new combinations and experiments, and also invites to reflect on the concept of networks, network structures, and the position of the involved actors.

Chloé

A project which especially emphasized the interaction aspect was presented on June 2, 2015 at the Jardin du Palais-Royal in Paris as part of a concert series during the Fete de la Musique. The concert was played by the DJane Chloé who used a system which was developed in collaboration with the Équipe de recherche IRCAM. The concept of the concert was to create a live interactive experience for the audience. Every visitor had the possibility to participate in the concert via smartphone.[12]

The stage was equipped with additional Wifi transmitters to provide enough internet service for the audience. Via smartphone browser, one could register online and log onto a web-socket where the position in relation to the stage had to be assigned. Once logged in, it appeared a graphical interface with interactive buttons and short playing instructions. The produced sound was played back via the loudspeakers of the individual smartphone.

Chloé could play her DJ set, but also control the sounds that were accessible by the audience. The initial position which was assigned by the smartphone user allowed her, to distribute the sounds in the audience space.

The new technology was used to integrate new forms of interactivity and combined already known concepts such as Dj-ing on stage, collaborative online platform, and smartphone gesture control with a concert situation. The audience was set in a double-position: spectator and interactive collaborator.

Still, this setting revealed some new challenges: In despite of installing an extra Wifi and the use of a generally sufficient technology, the connection was not stable enough for the envisaged number of participants. The use of the interface was easy, but not completely intuitive on the first try. Especially the fact that the sound would be played back by the internal loudspeaker of the smartphone was hard to discover due to the high volume of the main stage and the surrounding audience. If the smartphone lost the Wifi connection, it would still show the last loaded instruction for a while without being working.

This hints at a very interesting point to discuss: To what extent is it possible and/or necessary for the active audience to retrace the interactions and understand the degree of interactivity. At the concert, it was for example not obvious that Chloé could distribute the sounds in space and how the relationship between the single player and the surrounding participants was.

5.2 Performance and Causality

There already exists an ongoing discussion on the possibilities of identifying causalities within performance systems and the potential need to connect those with physical or virtual visible actions. Julian Rohrhuber stated that

> "Because a computer music network usually includes various active participants (people, processes) that are spread all over space but are all potentially connected in the most unusual ways, causation becomes a really interesting issue both for audience and for the musicians."[13, p. 150]

Increased by the separation of sound source and sound output device as well as latency, the action of a performer and the resulting consequence is no longer necessarily obvious. It becomes impossible for the audience to understand if, following Rohrhuber, an action was „random (could as well be otherwise), consequential (due to a rule), or intentional (aiming at something)".[13]

Direct communication like eye contact, facial expression, body language, or the commonly experienced sound can help to reveal invisible connections. In networks covering bigger distances, these issues are difficult to trace. Connected performer may be invisible for their co-musicians and the audience, and the experienced sound can differ in timing and quality for each linked performer. It also becomes evident that, as playing music on a computer does not necessarily involve physical movements, the lack of bodily expression again enhances the difficulty to retrace actions and reactions even if the performers are playing on stage for their audience.

6. PERFORMING COMPUTER NETWORK MUSIC

With the constantly improving network technologies, also the technologies used in Computer Network Music projects change. These may solve previous insufficiencies, but also offer new performing aspects, or foster new artistic ideas, which again may face some of the already familiar challenges. Especially important seems the traceability of musical interaction and the impression of the music being played live. These aspects open up a wide range of related projects. In the following, these challenges serve as the starting point for some brief reflections on performing Computer Network Music.

Comprehensibility

Understanding the causalities within the performance can be very important for the audience's experience. One possibility to clarify the ongoing actions for the audience is by viewing them. There exists a huge variety of visualizations, e.g. showing performers' screens, a compilation of their visual communication level, notation, code parts, visualizations of the sound or the sound production, as well as more abstract images or video clips underlining structures or enhancing the intended mood. While the last mentioned examples do not always aim at clarifying the performances' structures, displaying notation is usually used to demonstrate musical processes.

Hajdu implemented a viewer add-on in quintet.net which could be connected to a projector for the audience. [11] In the performance of his composition *Ivresse '84*, there was displayed the constantly changing notation as seen by the musicians, together with text excerpts from an interview Hajdu held with Janòs Négyesy about his first performance of Cage's *Freeman Etudes*. The text used the font "Cage", an adaption of John Cage's handwriting.[14]

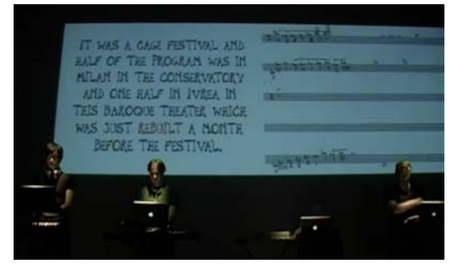


Figure 1. Screenshot from the rehearsal-video of *Ivresse '84* by the European Bridge Ensemble at Mucsarnok Budapest in September 2007.[15]

A similar visual idea was presented by Kingsley Ash and Nikos Stavropoulos. In their *Livecell project*, notation for a string quartet was generated by stochastic processes based on cellular automata which were set in real-time by the conductor on his screen.[16] The audience could see the conductor's screen with the cellular automata and

their developments as well as a view of the resulting notation played back live by the string quartet.[17]

In both examples, it was impossible to understand the causalities between the action on the computer and the emerging notation by just experiencing the performance. I n *Ivresse '84*, it was also impossible to allocate what sound each single computer musicians was playing. However, displaying the notation seemed sufficient enough for the audience to accept that there was a causality within the performance. The impression of musical interaction was fostered by the live playing musicians. But also being able to experience people reacting during the performance supported this impression, e.g. in the *Livecell project,* the conductor was standing almost within the audience. This in combination with seeing him setting the cellular automata on the screen helped to understand the artistic idea, as using the "Cage"-font always reminded on the content of the composition.

In computer music, notation can also include code. Viewing code is known from live-coding performances, where the projection of the live written code or forms of this code has been established by now, and has also been already discussed concerning aesthetic questions.[1] The visualization may reveal the ongoing processes directly through the live code, in more abstract terms, or open up interpretative dimensions. This developments did also directly influence Computer Network Music performances. For example the duo *Canute,* who combined an electronic drum set played by Matthew Yee-King, and live code by Alex McLean.[18] In their performance in 2014, both musicians played on stage, projecting code on the screen behind the stage – and on top of them. The projected code got now and then distorted, giving the impression of an interaction between the projection and the ongoing sound.[19] In this case, the visualization did not give any information about the compositional idea, nor did it clearly outline the ongoing processes. Instead, in presenting the live code with visual effects, the live character and the audience's experience of the concert as one interactive real-time event seems to come to the fore.

Live

One of the big challenges in Computer Network Music is to make understandable that the audience can experience live music. Besides displaying the code for the audience in real-time at live coding performances, matching visuals and light shows underlining or illustrating the sound, are one option to create the impression of a live performance.

Another possibility to proof that the music is played live is an interaction with the audience. Depending on the performance system, it may not always be clear to the audience if the audible output is the result of their real-time interactions. Therefore, interaction with the audience requires a clear assignment of the audience's function. If the interaction is on technical level, the causalities have to be clearly explained. Else, the actions seem to be unsigned, which can lead to a loss of interest or, even worse, to frustration for the audience. As previously outlined, in case of Chloé's performance, a lot of audience interaction took part even though the whole structure of the system was not completely clear to everyone. Technical problems with the web-socket distracted from listening the concert. But as the performance's sound output always seemed sufficient, the audience got the feeling of participating and contributing to the success of the performance, even though their own interaction would effectively not work. Here, the fact that they had the option to participate was enough to create the feeling of participation and therefore accept the fact that the music was played live. Chloé on the other hand accompanied her actions on the controllers by moving her body in the rhythm of the beat, and guided the audience's interaction by wide gestures like a conductor.

In contrast to the effort of showing that the music is played in realtime and creating a comprehensible "live music"-feeling for the audience, other performances do expose the minimalistic musician-computer-interaction. Even though these performances are designed for a physically present audience, the performers do not endeavor a stage show but concentrate on the pure sound experience. This may derive from a new interest in self-containing LAN-systems structured almost like chamber ensembles. [20]

Figure 2. Picture from the concert of the Birmingham Ensemble for Electroacoustic at Network Music Festival 2014 in Birmingham.[21]

The Birmingham Ensemble for Electroacoustic Research (BEER) performed at the Network Music Festival in 2014 with the three members sitting at a table surrounded by the audience. The live concert feeling was created by the light situation: the audience was seated in the dark while the musician's table was lighted from above.

In the same year, John Bischoff, Chris Brown and Tim Perkis performed at Active Music Festival in Oakland, each sitting behind a small table with the laptop in dimmed light on the stage facing the audience. This rather static set-up allowed the audience to see the hands of the performer and therefore see when they pressed or turned knobs of their controllers. Additionally, it was possible to see the attention switching between laptop screen and controller. Even though there was no visible interaction by eye contact or body language between the performer, and the actions of the performer could hardly be assigned to a certain sound event, the close inspection of the performers' activity were enough to get the feeling of the music being played live.

Figure 3. Screenshot from set Bischoff, Brown, Perkis at Active Music Festival, Oakland.[22]

7. SUMMARY

Using new technologies may solve technical insufficiencies and offer new performing aspects, but as outlined, it does not automatically solve performative challenges and may also rise a set of new questions.

One point of discussion is the impact of networked authorship on the aesthetics of live music making. Does the fact that The Hub is playing influence our perception?

Not discussed was also the role of the network system for the project's performance. How do we deal with the use of hierarchic networks in live music projects? What are the features that can make a performance system unique? How can this be transported within the performances?

For Computer Network Music, a successful performance can depend on very divers criteria. Performing this kind of music live on stage will always rise questions concerning the need on assignable structures or traceable causalities. At the same time, there exist many possibilities to create a situation for the audience which promotes the live character of the musical performance.

REFERENCES

[1] S. Gresham-Lancaster, "COMPUTER MUSIC NETWORK", *Proceedings of the Arts, Humanities, and Complex Networks – 4th Leonardo Satellite Symposium NetSci2013*, Berkeley, 2013, kindle edition, n.pag.

[2] Art. „network", in: International Electrotechnical Commission (Ed.), *Electricity, Electronics and Telecommunications. Multilingual Dictionary, Amsterdam/New York, Elsevier, 1992, pp.* 537-541.

[3] G. Föllmer, *Netzmusik*, Hofheim, Wolke, 2005.

[4] P. Manning, *Electronic and Computer Music,* Oxford, Oxford University Press, 2013.

[5] G. Weinberg, "Interconnected Musical Networks: Toward a Theoretical Framework", *Computer Music Journal*, 29(2), 2005, pp. 23-39.

[6] W. Duckworth, *Virtual Music. How the Web Got Wired for Sound*, New York, Routledge, 2005.

[7] A. Hugill, "Internet music: An introduction", *Contemporary Music Review*, Vol. 24, No. 6, 2005, pp. 429-437.

[8] N. Collins, "Zwischen 'data' und 'date'. Erfahrungen mit Proto-Web-Musik von The Hub", *Positionen*, Vol. 31, pp. 20-22.

[9] A. Carôt, A. Renaud and P. Rebelo, "Networked Music Performances: State Of The Art.", *Proceedings of the AES 30th International Conference*, Saariselkä 2007, pp. 131-137.

[10] C. Chafe, "CCRMA SOUND Wire", retrieved from https://ccrma.stanford.edu/groups/soundwire/ on Feb. 28, 2016.

[11] G. Hajdu, "Quintet.net – A Quintet on the Internet", *Proceedings of the International Computer Music Conference ICMC*, Singapore 2003, pp. 315-318.

[12] IRCAM (Ed.), "Chloé x IRCAM", retrieved from http://manifeste2015.ircam.fr/events/event/chloe-meets-ircam/ on Feb. 28, 2016.

[13] J. Rohrhuber, "Network music", in: Nicolas Collins (Ed.), *The Cambridge Companion to Electronic Music*, Cambridge University Press, Cambridge 2007, pp. 140-155.

[14] G. Hajdu, "Playing Performers: Ideas about Mediated Network Music Performance", *Proceedings of the Music in the Global Village Conference,* Budapest 2007, pp. 41-42.

[15] idem, *Ivresse '84*, Screenshot from the rehearsal of the European Bridge Ensemble EBE at Mucsarnok Budapest in September 2007, retrieved from www.youtube.com/watch?v=4TNrO871k-Y on May 19, 2016.

[16] A. Kingsley and N. Stavropoulos, "Stochastic processes in the musification of cellular automata: A case study of the Livecell project", *Emille Journal*, Vol. 10, Seoul 2012, pp. 13-20.

[17] idem, "Livecell project", Demonstration during the Korean Electro-Acoustic Music Conference on Oct. 27, 2012, Seoul.

[18] S. Knotts (Ed.), "Canute", Program of the Network Music Festival 2014, retrieved from http://network-musicfestival.org/nmf2014/programme/performances/canute/ on May 19, 2016.

[19] A. McLean, "Canute live in Jubez Karsruhe Algorave", Performance in Karlsruhe at January 17, 2015, retrieved from www.youtube.com/watch?v=uAq4BAbvRS4 and http://canute.lurk.org/ on May 19, 2016.

[1] See for example R. Bell, "Considering Interaction in Live Coding through a Pragmatic Aesthetic Theory", eContact!, Vol. 16.2, Montreal 2014, retrieved from http://econtact.ca/16_2/bell_livecoding.html on May 19, 2016.

[20] M. Akkermann, “Computer Network Music. Approximation to a far scattered history”, *Proceedings of the EMS,* Berlin 2014, retrieved from www.ems-network.org/spip.php?article363 on May 19, 2016.

[21] S. Knotts (Ed.), “BEER”, Program of the Network Music Festival 2014, retrieved from http://network-musicfestival.org/nmf2014/programme/performances/beer/ on May 19, 2016.

[22] anon., “Bischoff Brown Perkis at the Active Music Festival (Excerpt)”, Screenshot from the retrieved from the set Bischoff, Brown, Perkis at Active Music Festival, Oakland 2014, www.youtube.com/watch?v=1qkxreCGj3Y on May 19, 2016.

The Emotional Characteristics of Mallet Percussion Instruments with Different Pitches and Mallet Hardness

Chuck-jee Chau and Andrew Horner
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
chuckjee@cse.ust.hk, horner@cs.ust.hk

ABSTRACT

The mallet percussion instruments are gaining attention in modern music arrangements, especially in movie soundtracks and synthesized furniture music. Using a combination of these instruments, a reasonable variety of timbres and pitches are possible. Players can further choose mallets of different makes, which in turn significantly alter the timbre of the sound produced. How can these sounds be used to suggest various moods and emotions in musical compositions? This study compares the sounds of the marimba, vibraphone, xylophone, and glockenspiel, with different mallet hardness over various pitch registers, and ten emotional categories: Happy, Heroic, Romantic, Comic, Calm, Mysterious, Shy, Angry, Scary, and Sad. The results show that the emotional category Shy increases with pitch while Sad decreases with pitch. High-valence categories generally have an arching curve with peak in the mid-range. Mysterious and Angry are least affected by pitch. The results also show that the vibraphone is the most emotionally diverse, having a high rank in the high-arousal categories. Mallet hardness matters more in low-arousal categories, in which marimba is ranked significantly higher.

1. INTRODUCTION

Recent research has found that different kind of musical instrument sounds have strong emotional characteristics, including sustained [1, 2] and non-sustained [3] instruments. For example, it has found that the trumpet, clarinet, and violin are relatively joyful compared to other sustained instruments, even in isolated sounds apart from musical context, while the horn is relatively sad. The marimba and xylophone are relatively happier compared to other non-sustained instruments, while the harp and guitar are relatively depressed.

Our recent work has considered the non-sustained instrument sounds [3] including the plucked violin, guitar, harp, marimba, xylophone, vibraphone, piano, and harpsichord. The marimba, xylophone, and vibraphone were ranked higher for emotional categories Happy and Heroic. The study has however only included single mid-range pitches of the instruments with equalized loudness for consistent comparison. Given the wide pitch range of these instruments, we were curious how the emotional characteristics would be affected. Would they behave similar to that of the piano [4]? Or will they be different due to the distinctions in the instruments? The piano study found that pitch had a strong effect on all the tested categories. High-valence categories increased with pitch but decreased at the highest pitches. Angry and Sad decreased with pitch. Scary was strong in the extreme low and high registers.

A unique feature for mallet percussion instruments is the possibility of mallets. Even with the same instruments, a distinctive choice of mallets can create a vast difference in timbre. Composers often indicate whether hard or soft mallets are to be used for a piece or a particular musical passage. How does mallet hardness affect the emotional characteristics of the instrument sounds?

The current study is formulated to measure the emotional characteristics of pitches with different mallet hardness on the marimba, xylophone, vibraphone, and glockenspiel, four common instruments in contemporary instrumental music. We have included representative pitches from the lowest on the marimba, to the highest on the glockenspiel. The basic mallet hardness included are hard and soft. They were compared pairwise over ten emotional categories: Happy, Heroic, Romantic, Comic, Calm, Mysterious, Shy, Angry, Scary, and Sad.

This work provides a systematic overview of the emotional characteristics of the mallet percussion instruments across the different octaves with different mallet hardness. The findings are of potential interest to composers, percussionists, and audio engineers.

2. BACKGROUND

Much work has been done on emotion recognition in music, and recent work has considered the relationship between emotion and timbre. Researchers have gradually established connections between music emotion and timbre. Scherer and Oshinsky [5] found that timbre is a salient factor in the rating of synthetic sounds. Peretz *et al.* [6] showed that timbre speeds up discrimination of emotion categories. They found that listeners can discriminate between happy and sad musical excerpts lasting only 0.25s, where factors other than timbre might not come into play. Eerola *et al.* [1] showed a direct connection between music emotion and timbre. The study confirmed strong correlations between features such as attack time and brightness and the emotion dimensions valence and arousal for one-second isolated instrument sounds.

We followed up Eerola’s work with our own studies of music emotion and timbre [2, 3, 4, 7] to find out if some sounds were consistently perceived as being happier or sadder in pairwise comparisons. We designed listening tests to compare sounds from various string, wind, and percussion instruments. The results showed strong emotional characteristics for each instrument. We ranked the instruments by the number of positive votes they received for each emotion category, and derived scale values using the Bradley–Terry–Luce (BTL) statistical model [8, 9]. The rankings and BTL values for correlated emotion categories were similar (e.g., Sad, Depressed, and Shy). The horn and flute were highly ranked for Sad, while the violin, trumpet, and clarinet were highly ranked for Happy. The oboe was ranked in the middle. In another experiment, the harp, guitar and plucked violin were highly ranked for Sad, while the marimba, xylophone, and vibraphone were highly ranked for Happy. And piano was ranked in the

middle.

Within one single instrument, pitch and dynamics are also essential to shape emotional characteristics. Krumhansl [10] investigated changes in emotion for subjects listening to three minute musical excerpts and found that large variations in dynamics and pitch resulted in significantly higher ratings for the category Fear. Work by Huron et al. [11] into the perception of sine tone and MIDI synthesized piano melodies found that higher-pitched melodies were considered more submissive than lower-pitched melodies. Different pitch and dynamics of isolated piano sounds are found to produce different emotional impressions [4]. For mallet percussion instruments, the choice of mallets gives another dimension. Freed [12] found that listening subjects can easily perceive the change in mallet hardness when hitting metal objects.

3. EXPERIMENT

Our experiment was a listening test, where subjects compared pairs of instrument sounds over different emotional categories.

3.1 Test Materials

3.1.1 Stimuli

The stimuli used in the listening tests were sounds of mallet percussion instruments with different combinations of pitch and mallet hardness. The instruments used were marimba, vibraphone, xylophone, and glockenspiel.

All sounds were from the RWC and Prosonus sample libraries. There were two sets of recordings, one being played with hard mallets and the other soft. To avoid the effect of intervals of pitches interfering the experiment results, we chose only the C pitches (C3–C8), with C3 the lowest and C8 the highest (as shown in Figure 1). There were 30 samples in total. All sounds used a 44,100 Hz sampling rate.

Any silence before the onset of each sound was removed. The sound durations were then truncated to 0.9 second using a 150 ms linear fade-out before the end of each sound. In all cases, the fade-outs sounded like a natural damping or release of the sound.

3.1.2 Emotional Categories

The subjects compared the stimuli in terms of ten emotional categories: Happy, Heroic, Romantic, Comic, Calm, Mysterious, Shy, Angry, Scary, and Sad. Like many previous studies, we included representatives of the four quadrants of the Valence–Arousal plane [13] (Happy, Sad, Angry, Calm), along with some others. When picking these ten emotional categories, we particularly had dramatic musical genres such as opera and musicals in mind, where there are typically heroes, villains, and comic-relief characters with music specifically representing each. The emotional characteristics in these genres are generally more obvious and less abstract than in pure orchestral music.

We chose to use simple English emotional categories so that they would be familiar and self-apparent to non-native English speakers, which are similar to Italian music expression marks traditionally used by classical composers to specify the character of the music. These emotional categories also provide easy comparison with the results of our previous work.

3.2 Test Procedure

There were 28 subjects hired for the listening test, with an average age of 21.3 (ranging from 19 to 25). All subjects were undergraduate students at our university. None of them reported any hearing problems.

The subjects were seated in a "quiet room", with residual noise mostly only due to computers and air conditioning. The noise level was further reduced with headphones. Sound signals were converted to analog with a Sound Blaster X-Fi Xtreme Audio sound card, and then presented through Sony MDR-7506 headphones.The subjects were provided with an instruction sheet containing definitions [14] of the ten emotional categories.

Every subject made pairwise comparisons on a computer among all the 30 combinations of pitches and mallet hardness for each emotional category. During each trial, subjects heard a pair of sounds from different instruments and were prompted to choose the sound that represented the given emotional category more strongly. Each combination of two different instruments was presented once for each emotional category, and the listening test totaled $\binom{30}{2}$ *combinations* $\times$ 10 *emotional categories* $=$ 4350 trials.

The listening test took about 3 hours, with a short break after every 30 minutes to help minimize listener fatigue and maintain consistency.

3.3 Analysis Procedure

The Bradley–Terry–Luce (BTL) model was used to derive rankings based on the number of positive votes each sound received for each emotional category. For each emotional category, the BTL scale values for all the combinations of dynamic and pitch sum up to 1. The BTL value for each sound is the probability that listeners will choose that sound when considering a certain emotional category. For example, if all 30 combinations were judged equally happy, the BTL scale values would be $1/30 = 0.033$. The 95% confidence intervals of the BTL values were obtained to test the significance of the instrument ranks.

4. EXPERIMENT RESULTS

The raw results were votes for each sound pair and each emotional category. Figure 2 displays the BTL scale values of the sounds, with the corresponding 95% confidence intervals. There are some obvious trends in the charts. The instruments are well-contrasted for Happy and Sad. The emotion distinctiveness for Scary and Mysterious is lower.

Based on the results in Figure 2, Figure 3 shows how often each tested sound was statistically significantly greater than the others (i.e., the bottom of its 95% confidence interval was greater than the top of the 95% confidence interval of the others).

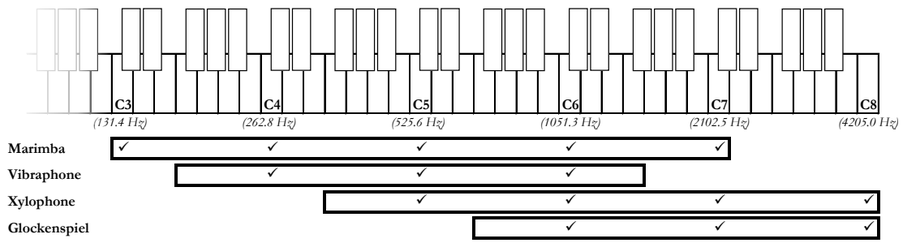


Figure 1: Selected pitches and the corresponding frequencies on the mallet percussion instruments. The concert pitch (as perceived) is used. *(A4=442Hz)*

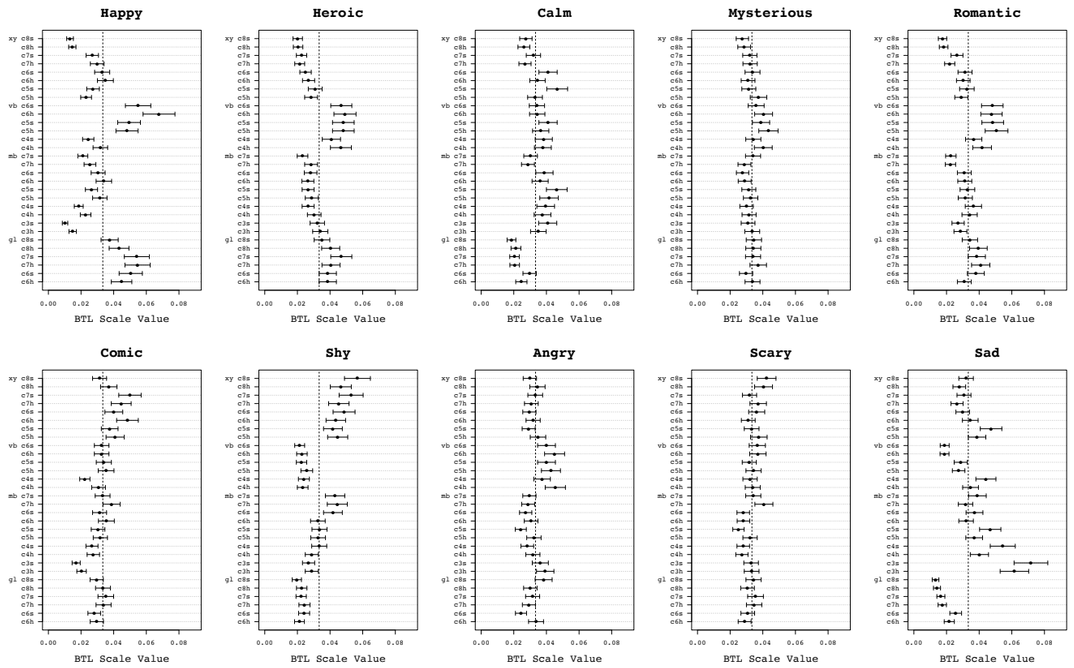


Figure 2: BTL scale values and the corresponding 95% confidence intervals. The dotted line represents no preference. `xy` = xylophone, `vb` = vibraphone, `mb` = marimba, `gl` = glockenspiel; `h` = hard, `s` = soft

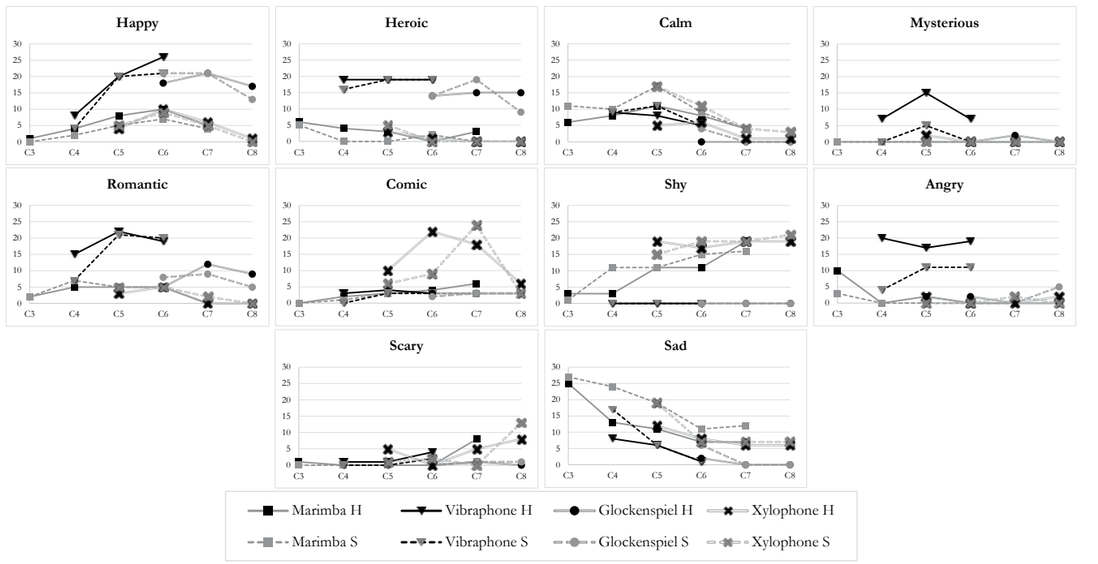


Figure 3: How often each sound was significantly greater than the others (i.e. the lower bound of its 95% confidence interval was greater than the upper bounds of their 95% confidence intervals). Since the number of instrument sounds is 30, the maximum possible value is 29. *(H = hard mallet (solid line), S = soft mallet (dotted line))*

For the emotional category Happy, Figure 3 shows a clear arching curve in pitch, with the peak at C6. The vibraphone and glockenspiel were more Happy than the marimba and xylophone. Mallet hardness did not have a very strong effect for Happy. The results for Romantic was similar, but the glockenspiel was less, and the peak was at C5 instead.

For Heroic, the figure shows again a distinct distance between vibraphone/glockenspiel, and marimba/xylophone. The differences in pitch and mallet hardness, however, had relatively little effect.

For Comic, the results show heightened responses for the xylophone. The difference for mallet hardness was more noteworthy. The hard mallet sounds had an arch peaking at C6, while the soft mallet sounds had an arch peaking at C7.

For Calm, the figure shows consistently calmer responses for soft mallet sounds. The strongest response was at C5. The marimba and xylophone were more Calm than the vibraphone and glockenspiel which were ranked at the bottom. Like Calm, soft mallet sounds were generally more Shy and more Sad than hard mallet sounds for marimba and xylophone. There was however a clear upward trend in pitch for Shy, and a clear downward trend with pitch for Sad.

Although Mysterious seems to show very little distinctiveness in Figure 2, Figure 3 shows that the vibraphone stood out somewhat among the tested instruments. The hard vibraphone sounds were much more Mysterious than the soft ones. In both cases, C5 was the peak. The vibraphone also stood out for Angry. Pitch difference had little effect though.

For Scary, the curves bottomed out across the range, with the exception of the highest pitches. The higher register of each instrument was usually scariest. The xylophone and marimba were considerably more Scary in the high register.

5. DISCUSSION

All ten emotional categories showed heightened responses for one or two of the four mallet percussion instruments. Nearly all emotional categories were strongly affected by pitch, with the possible exception of Heroic. Scary was also relatively unaffected by pitch except the highest register. Six of the emotional categories showed some strong effects due to mallet hardness. Happy, Heroic, Romantic, and Shy were less affected by mallet hardness.

The vibraphone had the most versatile character among the tested instruments. It was strongest for Happy, Heroic, Romantic, and also Mysterious and Angry. The glockenspiel was strongest for Happy and Heroic. These are generally the high-Arousal categories. The marimba and xylophone were strongest for Calm, Shy, and Sad. In addition, the xylophone was also very Comic and a little bit Scary. Except Scary, these are the low-Arousal categories.

The pitch trends were relatively clear. Compared with our previous study of pitch and dynamics on the piano [4], several emotional categories showed similar trends. Mid-high pitches were regarded as more Happy, Romantic, Comic, and Shy. Low pitches were Sad. High pitches were Scary. There were some contrasts though. On the piano, high pitches were most Calm and Mysterious.

Despite the limited availability of different types of mallets in sound library recordings, mallet hardness shows a strong effect on the emotional categories Comic, Mysterious, Angry, and Sad. The hard mallet sounds on the vibraphone were uniquely both Mysterious and Angry. The soft mallet sounds were Calm and Sad. Surprisingly, the soft mallet sounds on the xylophone shift the peak to a higher pitch for Comic.

The results confirm some existing common practices for emotional emphasis (e.g., using the xylophone for ragtime or lighthearted music, or using the vibraphone for romantic jazz). However, they also identify some less-commonly understood characteristics of the mallet percussion instruments, such as the Mysterious and Angry quality of vibraphone. Further timbral analysis of the experiment results will give more insights about the emotional characteristics of mallet percussion sounds. This will give recording and audio engineers, composers, and percussionists an interesting perspective on the emotional range of the instruments.

6. REFERENCES

[1] T. Eerola, R. Ferrer, and V. Alluri, "Timbre and affect dimensions: evidence from affect and similarity ratings and acoustic correlates of isolated instrument sounds," *Music Perception*, vol. 30, no. 1, pp. 49–70, 2012.

[2] B. Wu, A. Horner, and C. Lee, "The correspondence of music emotion and timbre in sustained musical instrument sounds," *J. Audio Eng. Soc.*, vol. 62, no. 10, pp. 663–675, 2014.

[3] C.-j. Chau, B. Wu, and A. Horner, "The emotional characteristics and timbre of nonsustaining instrument sounds," *J. Audio Eng. Soc*, vol. 63, no. 4, pp. 228–244, 2015.

[4] C.-j. Chau and A. Horner, "The effects of pitch and dynamics on the emotional characteristics of piano sounds." in *Proc. 41st Int. Comp. Music Conf. (ICMC)*, 2015, pp. 372–375.

[5] K. R. Scherer and J. S. Oshinsky, "Cue utilization in emotion attribution from auditory stimuli," *Motivation and Emotion*, vol. 1, no. 4, pp. 331–346, 1977.

[6] I. Peretz, L. Gagnon, and B. Bouchard, "Music and emotion: perceptual determinants, immediacy, and isolation after brain damage," *Cognition*, vol. 68, no. 2, pp. 111–141, 1998.

[7] R. Mo, B. Wu, and A. Horner, "The effects of reverberation on the emotional characteristics of musical instruments," *J. Audio Eng. Soc*, vol. 63, no. 12, pp. 966–979, 2015.

[8] R. A. Bradley, "Paired comparisons: Some basic procedures and examples," *Nonparametric Methods*, vol. 4, pp. 299–326, 1984.

[9] F. Wickelmaier and C. Schmid, "A Matlab function to estimate choice model parameters from paired-comparison data," *Behavior Research Methods, Instruments, and Computers*, vol. 36, no. 1, pp. 29–40, 2004.

[10] C. L. Krumhansl, "An exploratory study of musical emotions and psychophysiology." *Canadian J. Experimental Psychology/Revue canadienne de psychologie expérimentale*, vol. 51, no. 4, p. 336, 1997.

[11] D. Huron, D. Kinney, and K. Precoda, "Relation of pitch height to perception of dominance/submissiveness in musical passages," *Music Perception*, vol. 10, no. 1, pp. 83–92, 2000.

[12] D. J. Freed, "Auditory correlates of perceived mallet hardness for a set of recorded percussive sound events," *J. Acoustical Soc. of Amer.*, vol. 87, no. 1, pp. 311–322, 1990.

[13] J. A. Russell, "A circumplex model of affect." *J. personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[14] Cambridge University Press. Cambridge academic content dictionary. [Online]. Available: http://dictionary.cambridge.org/dictionary/american-english

The Effects of Pitch and Dynamics on the Emotional Characteristics of Bowed String Instruments

Samuel J. M. Gilburt
School of Computing Science
Newcastle University
Newcastle-upon-Tyne, NE1 7RU, United Kingdom
s.j.m.gilburt@ncl.ac.uk

Chuck-jee Chau, Andrew Horner
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
chuckjee@cse.ust.hk, horner@cs.ust.hk

ABSTRACT

Previous research has shown that different musical instrument sounds have strong emotional characteristics. This paper investigates how emotional characteristics vary with pitch and dynamics within the bowed string instrument family. We conducted listening tests to compare the effects of pitch and dynamics on the emotional characteristics of the violin, viola, cello, and double bass. Listeners compared the sounds pairwise over ten emotional categories. The results showed that the emotional characteristics Happy, Heroic, Romantic, Comic, and Calm generally increased with pitch, but decreased at the highest pitches. The characteristics Angry and Sad generally decreased with pitch. Scary was strong in the extreme low and high registers, while Shy was relatively unaffected by pitch. In terms of dynamics, the results showed that Heroic, Comic, and Angry were stronger for loud notes, while Romantic, Calm, Shy, and Sad were stronger for soft notes. Surprisingly, Scary was least affected by dynamics. These results provide audio engineers and musicians with possible suggestions for emphasizing various emotional characteristics of the bowed strings in sound recordings and performances.

1. INTRODUCTION

Previous research has shown that different musical instrument sounds have strong emotional characteristics [1, 2, 3, 4, 5, 6, 7]. For example, among sustained instruments the violin has been found to be happier in character than the horn, even isolated from any musical context. Going further, it would be interesting to know how emotional characteristics vary with pitch and dynamics within a family of instruments such as the bowed strings. A number of interesting questions arise:

Regarding pitch in the bowed strings:

- Which emotional characteristics tend to increase/decrease with increasing pitch?
- Are any emotional characteristics relatively unaffected by pitch?

Regarding dynamics in the bowed strings:

- Which emotional characteristics are stronger with loud/soft notes?
- Are any emotional characteristics relatively unaffected by dynamics?

Answering these questions will help quantify the emotional effects of pitch and dynamics in the bowed strings. The results will provide possible suggestions for musicians in orchestration, performers in blending and balancing instruments, and audio engineers in recording and mixing bowed string instruments.

2. BACKGROUND

Previous research has investigated emotion recognition in music, especially addressing melody [8], harmony [9, 10], rhythm [11, 12], lyrics [13], and localization cues [14]. Similarly, researchers have found timbre to be useful in a number of applications such as automatic music genre classification [15], automatic song segmentation [16], and song similarity computation [16]. Researchers have also considered music emotion and timbre together in a number of studies, which are reviewed below. We also review previous research on the timbre of bowed string instruments as well as the effects of pitch and dynamics in musical excerpts.

2.1 Music Emotion and Timbre

Hevner's early work [17] pioneered the use of adjective scales in music and emotion research. She divided 66 adjectives into 8 groups where adjectives in the same group were related and compatible. Scherer and Oshinsky [18] used a 3D dimensional model to study the relationship between emotional attributes and synthetic sounds by manipulating different acoustic parameters such as amplitude, pitch, envelope, and filter cut-off. Subjects rated sounds on a 10-point scale for the three dimensions Pleasantness, Activity, and Potency. They also allowed users to label sounds with emotional labels such as Anger, Fear, Boredom, Surprise, Happiness, Sadness, and Disgust.

Bigand *et al.* [19] conducted experiments to study emotion similarities between one-second musical excerpts. Hailstone *et al.* [20] studied the relationship between sound

identity and music emotion. They asked participants to select which one of four emotional categories (Happiness, Sadness, Fear, or Anger) was represented in 40 novel melodies that were recorded in different versions using electronic synthesizer, piano, violin, and trumpet, controlling for melody, tempo, and loudness between instruments. They found a significant interaction between instrument and emotion.

Eerola *et al.* [1] studied the correlation of perceived emotion with temporal and spectral sound features. They asked listeners to rate the perceived affect qualities of one-second instrument tones using 5 dimensions: Valence, Energy, Tension, Preference, and Intensity. Orchestral and some exotic instruments were included in their collection.

Asutay *et al.* [21] studied Valence and Arousal along with loudness and familiarity in subjects' responses to environmental and processed sounds. Subjects were asked to rate each sound on 9-point scales for Valence and Arousal. Subjects were also asked to rate how Annoying the sound was.

Wu *et al.* [2, 3, 4, 6] and Chau *et al.* [5, 7] compared the emotional characteristics of sustaining and non-sustaining instruments. They used a BTL model to rank paired comparisons of eight sounds. Wu compared sounds from eight wind and bowed string instruments such as the trumpet, flute, and bowed violin, while Chau compared eight non-sustaining sounds such as the piano, plucked violin, and marimba. Eight emotional categories for expressed emotion were tested including Happy, Sad, Heroic, Scary, Comic, Shy, Joyful, and Depressed. The results showed distinctive emotional characteristics for each instrument. Wu found that the timbral features spectral centroid and even/odd harmonic ratio were significantly correlated with emotional characteristics for sustaining instruments. Chau found that decay slope and density of significant harmonics were significantly correlated for non-sustaining instruments.

Perhaps most relevant to the current proposal in methodology, recently Chau and Horner [22] studied the emotional characteristics of the piano with varying pitch and dynamics. They found the emotional characteristics Happy, Romantic, Calm, Mysterious, and Shy generally increased with pitch, while Heroic, Angry, and Sad generally decreased with pitch on the piano. They also found that Comic was strongest in the mid-register, and Scary was strongest in the extreme low and high registers. For dynamics on the piano, they found Heroic, Comic, and Angry were stronger for loud notes, while Romantic, Calm, Shy, and Sad were stronger for soft notes, and Happy, Mysterious, and Scary were relatively unaffected by dynamics.

2.2 Bowed String Instruments

Various previous research has considered bowed string instrument sounds. Brown and Vaughn [23] used pairwise comparisons of one-second vibrato and non-vibrato violin tones, and found that the perceived pitch of vibrato tones was the mean of the variation. Krumhansl [24] empirically studied memorability, openness, and emotion in two string ensemble pieces, a Mozart string quintet and a Beethoven string quartet. She noted that subjects found it difficult to describe an overall emotional response to the pieces as a whole, suggesting that the wide range of contrasting emotions portrayed may have been responsible for the mixed response. So and Horner [25] investigated methods for synthesizing inharmonic bowed string tones. A wavetable matching technique was used to improve matches to inharmonic string tones.

2.3 Pitch and Dynamics

There has been some research into the effect of varying pitch and dynamics on musical excerpts. Kamenetsky *et al.* [26] investigated the effects of tempo and dynamics on the perception of four MIDI musical excerpts. One version had static tempo and dynamics, another varying tempos, another varying dynamics, and the last varying tempos and dynamics. Participants then rated each excerpt on a 7-point scale for likeability and emotional expressiveness. While tempo was found to have no effect on the ratings, variations in dynamics were found to result in higher ratings for both measurements. Krumhansl [27] investigated changes in emotion for subjects listening to three-minute musical excerpts and found that large variations in dynamics and pitch resulted in significantly higher ratings for the category Fear. Work by Huron *et al.* [28] into the perception of sine tone and MIDI synthesized piano melodies found that higher-pitched melodies were considered more submissive than lower-pitched melodies.

3. EXPERIMENT METHODOLOGY

We conducted listening tests to compare the effect of pitch and dynamics on the emotional characteristics of individual bowed string instrument sounds. We tested the violin, viola, cello, and double bass at three or four different pitches, and at both *forte* (loud) and *piano* (soft) dynamic levels. We compared the sounds pairwise over ten emotional categories (Happy, Heroic, Romantic, Comic, Calm, Mysterious, Shy, Angry, Scary, and Sad) to determine the effect of pitch and dynamics.

3.1 Stimuli

The experiment used sounds from the four main instruments in the Western bowed string family: violin (Vn), viola (Va), violoncello (Vc), and contrabass (Cb). The sounds were obtained from the Prosonus sample library [29]. The sounds presented were approximately 0.9 s in length. For each comparison, the first sound was played, followed by 0.2 s of silence, and then the second sound. Thus the total for one comparison was 2 s. The sounds for each instrument were as follows:

- Vn: C4, C5, C6, C7
- Va: C3, C4, C5
- Vc: C2, C3, C4, C5
- Cb: C1, C2, C3

The sounds were all C's of different octaves so as to avoid other musical intervals influencing the emotional responses of the subjects. Each note also had two dynamic variations, corresponding to *forte (f)* and *piano (p)*—loud and soft. The total number of sounds was 28 (14 notes $\times$ 2 dynamic levels).

The instrument sounds were analyzed using a phase-vocoder algorithm, where bin frequencies were aligned with harmonics [30]. Temporal equalization was carried out in the frequency domain, identifying attacks and decays by inspection of the time-domain amplitude-vs.-time envelopes. These envelopes were reinterpolated to achieve a standardized attack time of 0.07 s, sustain time of 0.36 s, and decay time of 0.43 s for all sounds. These values were chosen based on the average attack and decay times of the original sounds. As different attack and decay times are known to affect the emotional responses of subjects [1], equalizing avoids this potential factor. The stimuli were resynthesized from the time-varying harmonic data using the standard method of time-varying additive sine wave synthesis (oscillator method) with frequency deviations set to zero. The fundamental frequencies of the synthesized sounds were set to exact octaves.

3.2 Emotional Categories

The subjects compared the stimuli in terms of ten emotional categories: Happy, Heroic, Romantic, Comic, Calm, Mysterious, Shy, Angry, Scary, and Sad. We selected these categories because composers often use these terms in tempo and expression markings in their scores. We chose to use simple English emotional categories so that they would be familiar and self-apparent to subjects rather than Italian music expression markings traditionally used by classical composers to specify the character of the music. The chosen emotional categories and related Italian expression markings are listed in Table 1.

Emotional Category	Commonly-used Italian musical expression marks
Happy	*allegro, gustoso, gioioso, giocoso, contento*
Heroic	*eroico, grandioso, epico*
Romantic	*romantico, affetto, afectuoso, passionato*
Comic	*capriccio, ridicolosamente, spiritoso, comico, buffo*
Calm	*calmato, tranquillo, pacato, placabile, sereno*
Mysterious	*misterioso, misteriosamente*
Shy	*timido, riservato, timoroso*
Angry	*adirato, stizzito, furioso, feroce, irato*
Scary	*sinistro, terribile, allarmante, feroce, furioso*
Sad	*dolore, lacrimoso, lagrimoso, mesto, triste*

Table 1. The ten chosen emotional categories and related music expression markings commonly used by classical composers.

One advantage of using a categorical instead of a dimensional emotional model is that it allows faster decision making by listening test subjects. However, these emotional categories can still be represented in a dimensional model, such as the Valence-Arousal model [31]. Their ratings according to the *Affective Norms for English Words* [32] are shown in Figure 1 using the Valence–Arousal model. Valence indicates the positivity of an emotional category; Arousal indicates the energy level of an emotional category. Though Scary and Angry are similar in terms of Valence and Arousal, they have distinctly different meanings. Likewise with Romantic, Happy, Comic, and Heroic.

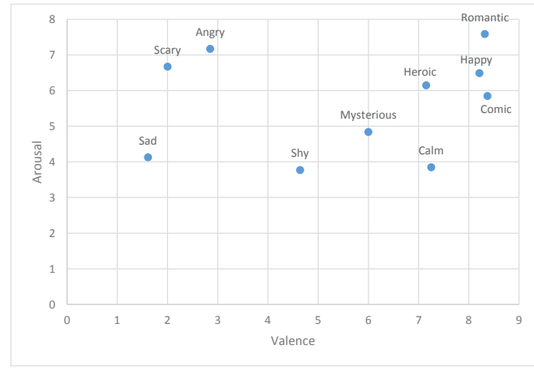


Figure 1. The distribution of the emotional characteristics in the dimensions Valence and Arousal. The Valence and Arousal values are given by the 9-point rating in *ANEW* [32].

3.3 Subjects

23 subjects were hired to take the listening test. All subjects were fluent in English. They were all undergraduate students at our university.

3.4 Listening Test

Each subject made paired comparisons of all sounds. During each trial, subjects heard a pair of sounds and were prompted to choose which sound better represented a given emotional characteristic. The listening test consisted of 28 sounds (14 notes $\times$ 2 dynamic levels) pairwise compared for 10 emotional categories, i.e. $28C2 \times 10 = 3780$ trials. The overall trial presentation order was randomized, for each emotional category (i.e. all Happy comparisons in random order first, followed by all Heroic comparisons in random order second, etc.). The emotional categories were presented in sequence so as to avoid confusing and fatiguing the subjects. Consistency checks showed that subjects maintained their level of concentration throughout the duration of the test. The test took approximately 3 hours to complete, with 5 minute breaks every 30 minutes to help minimize listener fatigue and maintain consistency.

Before commencing the test, subjects read online definitions of the emotional categories from the Cambridge Academic Content Dictionary [33]. The subjects were seated in a quiet room with little background noise (39dB SPL), and used Sony MDR-7506 over-ear headphones which provided further noise reduction. Sound signals were converted to analog by a Sound Blaster X-Fi Xtreme Audio sound card and presented through the headphones at a volume level of 78dB SPL, as measured with a sound meter. The Sound Blaster DAC utilizes 24 bits with a maximum sampling rate of 96kHz and a 108dB S/N ratio.

4. RESULTS

We ranked the sounds by the number of positive votes received for each emotional category, deriving scale values using the Bradley–Terry–Luce (BTL) statistical model [34]. The BTL values for each emotional category sum to 1. The BTL value given to a sound is the probabil-

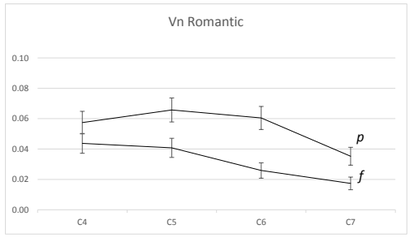


Figure 2. BTL scale values and the corresponding 95% confidence intervals for Romantic for violin.

ity that listeners will choose that sound when considering a given emotional category. For example, if all 28 sounds (14 notes $\times$ 2 dynamic levels) were considered equally Happy, the BTL scale values would be $1/28 \approx 0.0357$. Figure 2 shows an example graph for the BTL scale values and the corresponding 95% confidence intervals for Romantic for violin.

Based on the BTL results, Figure 3 shows how often each instrument sound was statistically significantly greater than the others (i.e., the bottom of its 95% confidence interval was greater than the top of the 95% confidence interval of the others).

For the emotional category Happy, Figure 3 shows that dynamics did not have a strong effect, except in the high register where soft notes were stronger. There was a clear upward trend in pitch, with the exception of the highest pitches of the violin and cello. The overall trend was an arching curve peaking at about C5, with the lowest pitches least Happy.

For Heroic, Figure 3 shows a strong response for loud notes across the middle and high registers (from C3 to C6). As with many of the high-Valence emotional categories, there were lower responses at extreme ends of the pitch range, with the lowest and highest notes least Heroic.

The results for Romantic show heightened responses for soft notes across all pitches and instruments. The overall trend was a similar arch to other high-Valence emotional categories such as Heroic, with mid-range pitches most Romantic.

For Comic, Figure 3 shows strong Comic responses for loud notes in the middle and high registers (though not the highest). Low pitches, especially those of the double bass, were least Comic. The high register of the loud cello was most Comic.

For Calm, Figure 3 shows consistently calmer responses for soft notes. The strongest response was in the mid-range, and the lowest pitches of the double bass and cello were least Calm.

Figure 3 shows that Mysterious varied little with pitch and dynamics, except the lowest pitches of the double bass where soft notes were more Mysterious.

Like Calm, soft notes were consistently more Shy than loud notes. Pitch had little effect across all instruments (though for loud notes, there was a slight downward trend with pitch).

For Angry, Figure 3 shows angrier responses to loud notes. There was a general downward trend with pitch. Consistent with this, the double bass and cello were more Angry, and the violin least Angry.

Scary was relatively unaffected by dynamics. The curves bottomed out across the entire mid-range, opposite to high-Valence emotions such as Heroic (in fact, Heroic and Scary are nearly mirror images of one another along the x-axis in Figure 3 for loud notes). The lowest and highest pitches were significantly scarier, a result which indirectly agrees with Krumhansl [27] who found that large variations in pitch resulted in significantly higher ratings for Fear. As a consequence, the double bass and violin were significantly more Scary than the cello and viola.

For Sad, soft notes were sadder. The lower register of each instrument was saddest, except the double bass. As expected, this contrasts with Happy where higher pitches were generally happier (in fact, Happy and Sad are nearly mirror images of one another along the y-axis in Figure 3).

5. DISCUSSION

Nine out of ten emotional categories were strongly affected by pitch (all but Shy), and Shy was relatively unaffected by pitch. Mysterious was also relatively unaffected by pitch except the double bass. Nearly all emotional categories showed some strong effects due to dynamics. Surprisingly, Scary was least affected by dynamics. Happy and Mysterious were also less affected by dynamics.

The results show that pitch generally had a similar effect on emotional categories with similar Valence. The high-Valence characteristics Happy, Heroic, Romantic, Comic, and Calm had broadly similar shapes in Figure 3 (mostly-increasing and arching), while the low-Valence characteristics Angry and Sad were decreasing. The middle-Valence characteristics, Mysterious and Shy, were less affected by pitch. Scary was the biggest exception, increasing with pitch rather than decreasing like the other low-Valence characteristics Angry and Sad. Dynamics had a similar effect on emotional categories with similar Arousal, though there were more exceptions. The high-Arousal characteristics Heroic, Comic, and Angry were strongest for loud notes, while the low-Arousal characteristics Calm, Shy, and Sad were strongest for soft notes. However, Romantic was opposite to this trend, and the high-Arousal categories Happy and Scary were relatively unaffected by dynamics.

The above results can give suggestions to musicians in orchestration, performers in blending and balancing instruments, and recording engineers in mixing recordings and live performances. Emotional characteristics can be manipulated in a recording, performance, or composition by emphasizing instruments, pitches, and dynamics that are comparatively stronger in representing these characteristics. The results confirm some existing common practices for emotional emphasis (e.g., using low double basses and high violins together for Scary passages). However, they also identify some less-commonly understood characteristics of the bowed strings, such as the Comic quality of the high cello at loud dynamics.

6. REFERENCES

[1] T. Eerola, R. Ferrer, and V. Alluri, “Timbre and affect dimensions: evidence from affect and similarity ratings and acoustic correlates of isolated instrument sounds,” *Music Perception*, vol. 30, no. 1, pp. 49–70, 2012.

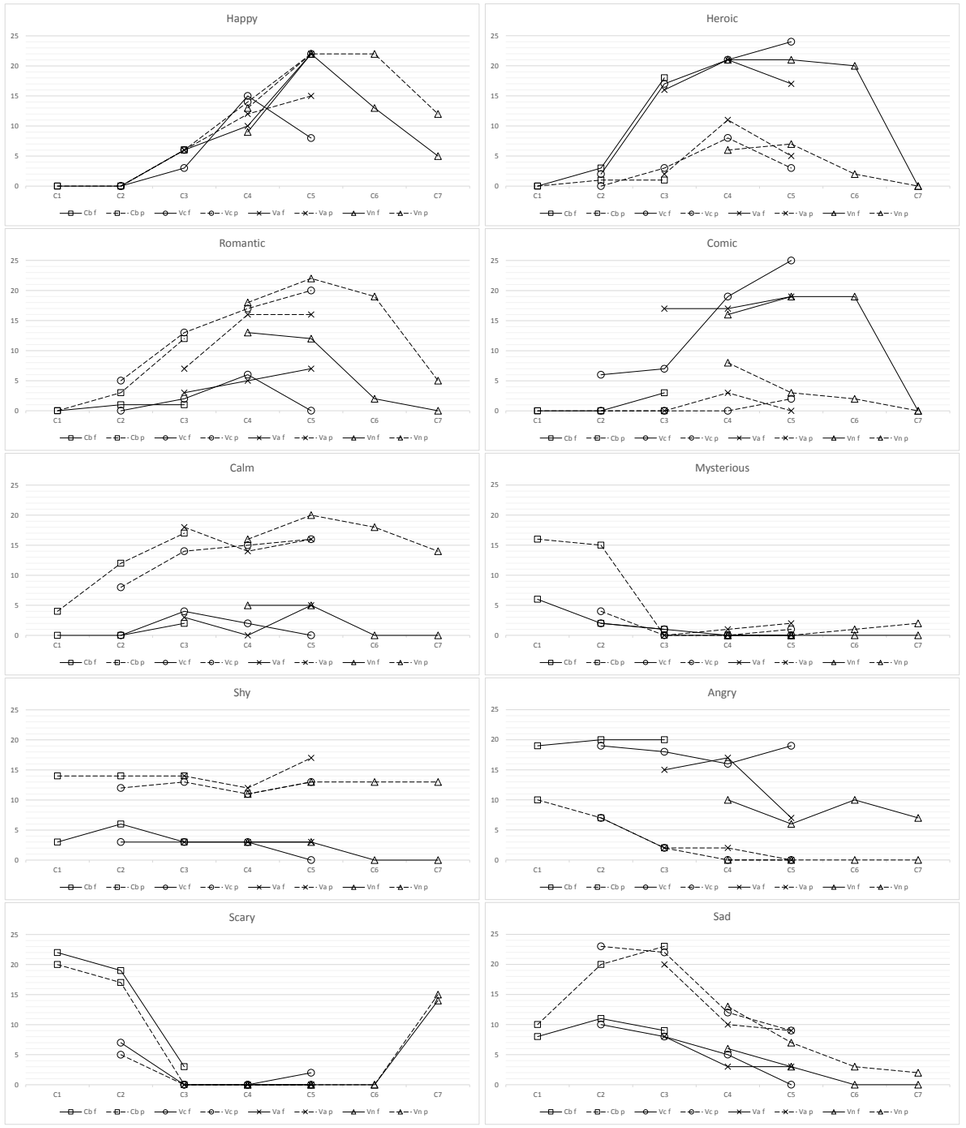


Figure 3. How often each instrument sound was significantly greater than the others (i.e. the lower bound of its 95% confidence interval was greater than the upper bounds of their 95% confidence intervals). Since the number of instrument sounds is 28, the maximum possible value is 27. Loud notes are connected by solid lines, and soft notes by dashed lines.

[2] B. Wu, S. Wun, C. Lee, and A. Horner, “Spectral Correlates in Emotion Labeling of Sustained Musical Instrument Tones,” in *Proc. 14th Int. Soc. Music Inform. Retrieval Conf. (ISMIR)*, November 4–8 2013.

[3] B. Wu, A. Horner, and C. Lee, “The Correspondence of Music Emotion and Timbre in Sustained Musical Instrument Sounds,” *J. Audio Eng. Soc.*, vol. 62, no. 10, pp. 663–675, 2014.

[4] ——, “Musical Timbre and Emotion: The Identification of Salient Timbral Features in Sustained Musical Instrument Tones Equalized in Attack Time and Spectral Centroid.” in *Proc. 40th Int. Comp. Music Conf. (ICMC)*, 2014, pp. 928–934.

[5] C.-j. Chau, B. Wu, and A. Horner, “Timbre Features and Music Emotion in Plucked String, Mallet Percussion, and Keyboard Tones.” in *Proc. 40th Int. Comp. Music Conf. (ICMC)*, 2014, pp. 982–989.

[6] B. Wu, A. Horner, and C. Lee, “Emotional Predisposition of Musical Instrument Timbres with Static Spec-

tra," in *Proc. 15th Int. Soc. Music Inform. Retrieval Conf. (ISMIR)*, 11 2014, pp. 253–258.

[7] C.-j. Chau, B. Wu, and A. Horner, "The Emotional Characteristics and Timbre of Nonsustaining Instrument Sounds," *J. Audio Eng. Soc*, vol. 63, no. 4, 2015.

[8] L.-L. Balkwill and W. F. Thompson, "A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues," *Music Perception*, vol. 17, no. 1, pp. 43–64, 1999.

[9] J. Liebetrau, J. Nowak, T. Sporer, M. Krause, M. Rekitt, and S. Schneider, "Paired Comparison as a Method for Measuring Emotions," in *Audio Eng. Soc. Convention 135*. Audio Eng. Soc., 2013.

[10] I. Lahdelma and T. Eerola, "Single chords convey distinct emotional qualities to both naïve and expert listeners," *Psychology of Music*, 2014.

[11] J. Skowronek, M. F. McKinney, and S. Van De Par, "A Demonstrator for Automatic Music Mood Estimation." in *Proc. Int. Soc. Music Inform. Retrieval Conf. (ISMIR)*, 2007, pp. 345–346.

[12] M. Plewa and B. Kostek, "A Study on Correlation between Tempo and Mood of Music," in *Audio Eng. Soc. Conv. 133*. Audio Eng. Soc., 2012.

[13] Y. Hu, X. Chen, and D. Yang, "Lyric-based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method." in *Proc. Int. Soc. Music Inform. Retrieval Conf. (ISMIR)*, 2009, pp. 123–128.

[14] I. Ekman and R. Kajastila, "Localization Cues Affect Emotional Judgments–Results from a User Study on Scary Sound," in *Audio Eng. Soc. Conf.: 35th Int. Conf.: Audio for Games*. Audio Eng. Soc., 2009.

[15] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, 2002.

[16] J.-J. Aucouturier, F. Pachet, and M. Sandler, ""The way it sounds": timbre models for analysis and retrieval of music signals," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1028–1035, 2005.

[17] K. Hevner, "Experimental studies of the elements of expression in music," *The American J. Psychology*, pp. 246–268, 1936.

[18] K. R. Scherer and J. S. Oshinsky, "Cue utilization in emotion attribution from auditory stimuli," *Motivation and Emotion*, vol. 1, no. 4, pp. 331–346, 1977.

[19] E. Bigand, S. Vieillard, F. Madurell, J. Marozeau, and A. Dacquet, "Multidimensional scaling of emotional responses to music: The effect of musical expertise and of the duration of the excerpts," *Cognition & Emotion*, vol. 19, no. 8, pp. 1113–1139, 2005.

[20] J. C. Hailstone, R. Omar, S. M. Henley, C. Frost, M. G. Kenward, and J. D. Warren, "It's not what you play, it's how you play it: Timbre affects perception of emotion in music," *The Quarterly J. Experimental Psychology*, vol. 62, no. 11, pp. 2141–2155, 2009.

[21] E. Asutay, D. Västfjäll, A. Tajadura-Jiménez, A. Genell, P. Bergman, and M. Kleiner, "Emoacoustics: A study of the psychoacoustical and psychological dimensions of emotional sound design," *J. Audio Eng. Soc.*, vol. 60, no. 1/2, pp. 21–28, 2012.

[22] C.-j. Chau and A. Horner, "The Effects of Pitch and Dynamics on the Emotional Characteristics of Piano Sounds." in *Proc. 41st Int. Comp. Music Conf. (ICMC)*, 2015.

[23] J. C. Brown and K. V. Vaughn, "Pitch center of stringed instrument vibrato tones," *J. Acoustical Soc. of Amer.*, vol. 100, no. 3, pp. 1728–1735, 1996.

[24] C. L. Krumhansl, "Topic in music: An empirical study of memorability, openness, and emotion in Mozart's String Quintet in C Major and Beethoven's String Quartet in A Minor," *Music Perception*, pp. 119–134, 1998.

[25] C. So and A. B. Horner, "Wavetable matching of inharmonic string tones," *J. the Audio Eng. Society*, vol. 50, no. 1/2, pp. 46–56, 2002.

[26] S. B. Kamenetsky, D. S. Hill, and S. E. Trehub, "Effect of tempo and dynamics on the perception of emotion in music," *Psychology of Music*, vol. 25, pp. 149–160, 1997.

[27] C. L. Krumhansl, "An exploratory study of musical emotions and psychophysiology." *Canadian J. Experimental Psychology/Revue canadienne de psychologie expérimentale*, vol. 51, no. 4, pp. 336–353, 1997.

[28] D. Huron, D. Kinney, and K. Precoda, "Relation of pitch height to perception of dominance/submissiveness in musical passages," *Music Perception*, vol. 10, no. 1, pp. 83–92, 2000.

[29] J. Rothstein, "ProSonus Studio Reference Disk and Sample Library Compact Disks," 1989.

[30] J. W. Beauchamp, "Analysis and synthesis of musical instrument sounds," in *Analysis, Synthesis, and Perception of musical sounds*. Springer, 2007, pp. 1–89.

[31] J. A. Russell, "A circumplex model of affect." *J. personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[32] M. M. Bradley and P. J. Lang, "Affective norms for English words (ANEW): Instruction manual and affective ratings," *Psychology*, no. C-1, pp. 1–45, 1999.

[33] Cambridge University Press. Cambridge Academic Content Dictionary. [Online]. Available: http://dictionary.cambridge.org/dictionary/american-english

[34] F. Wickelmaier and C. Schmid, "A Matlab function to estimate choice model parameters from paired-comparison data," *Behavior Research Methods, Instruments, and Computers*, vol. 36, no. 1, pp. 29–40, 2004.

The Effects of MP3 Compression on Emotional Characteristics

Ronald Mo
Department of Computer Science and Engineering,
Hong Kong University of Science and Technology,
Hong Kong
ronmo@cse.ust.hk

Ga Lam Choi
Department of Computer Science and Engineering,
Hong Kong University of Science and Technology,
Hong Kong
glchoi@cse.ust.hk

Chung Lee
The Information Systems Technology and Design Pillar,
Singapore University of Technology and Design,
20 Dover Drive, Singapore 138682
chung_lee@sutd.edu.sg

Andrew Horner
Department of Computer Science and Engineering,
Hong Kong University of Science and Technology,
Hong Kong
horner@cse.ust.hk

ABSTRACT

Previous research has shown that MP3 compression changes the similarities of musical instruments, while other research has shown that musical instrument sounds have strong emotional characteristics. This paper investigates the effect of MP3 compression on music emotion. We conducted listening tests to compare the effect of MP3 compression on the emotional characteristics of eight sustained instrument sounds. We compared the compressed sounds pairwise over ten emotional categories. The results show that MP3 compression strengthened the emotional characteristics Sad, Scary, Shy, and Mysterious, and weakened Happy, Heroic, Romantic, Comic, and Calm. Interestingly, Angry was relatively unaffected by MP3 compression.

1. INTRODUCTION

Though most listeners know that extreme MP3 compression degrades audio quality, many are willing to compromise quality for convenience. This is reflected to the current portable music consumption trend where consumers are using internet music streaming services more frequently than buying CDs or downloads [1]. Major streaming services use MP3 compression.

As previous research has shown that musical instrument sounds have strong and distinctive emotional characteristics [2, 3, 4, 5, 6], it would be interesting to know how MP3 compression affects the emotional characteristics of musical instruments. In particular, we will address the following questions: What are the emotional effects of MP3 compression? Do all emotional characteristics decrease about equally with more compression? Which emotional characteristics increase or decrease with more compression? Which emotional characteristics are unaffected by more compression? Which instruments change the most and least with more compression?

2. BACKGROUND

2.1 MP3 Compression

MP3 compression reduces the size of audio files by discarding less audible parts of the sound. When an instrument sound is encoded using an MP3 codec, due to the lossy nature of MP3 compression, the sound is altered. The perceptual quality of lossy compression is a longstanding subject of digital audio research. Zwicker found a number of characteristics of the human auditory system including simultaneous masking and temporal masking formed a part of the psychoacoustic model of MP3 encoders [7]. Van de Par and Kohlrausch proposed a number of methods to evaluate different audio compression codecs [8].

Various studies have investigated the perceptual artifacts generated by low bit rate audio codecs. Erne produced a CD-ROM that demonstrates some of the most common coding artifacts in low bit rate codecs. They explained and presented audio examples for each of the coding artifacts separately using different degrees of distortion [9]. Chang et al. constructed models of the audible artifacts generated by temporal noise shaping and spectral band replication, which are far more difficult to model using existing encoding systems [10]. Marins carried out a series of experiments aiming to identify the salient dimensions of the perceptual artifacts generated by low bit rate spatial audio codecs [11].

Previous studies have also subjectively evaluated the perceptual quality loss in MP3 compression [12, 13, 14, 15]. A recent study evaluated the discrimination of musical instrument tones after MP3 compression using various bit rates [16]. A following study [17] compared dissimilarity scores for instrument tone pairs after MP3 compression to determine whether instrument tones sound more or less similar after MP3 compression, and found that MP3 can change the timbre of musical instruments.

2.2 Music Emotion and Timbre

Researchers have considered music emotion and timbre together in a number of studies, which are well-summarized in [6].

3. METHODOLOGY

3.1 Overview

We conducted listening tests to compare pairs of original and compressed instrument sounds over different emotional categories. Paired comparisons were chosen for simplicity. This section gives further details about the listening test.

3.2 Listening Test

We used eight sustained instrument sounds: bassoon (bs), clarinet (cl), flute (fl), horn (hn), oboe (ob), saxophone (sx), trumpet (tp), and violin (vn). The sustained instruments are nearly harmonic, and the chosen sounds had fundamental frequencies close to Eb4 (311.1 Hz). All eight instrument sounds were also used by a number of other timbre studies [16, 17, 18, 19, 20, 21, 22, 23, 24]. Using the same samples makes it easier to compare results.

Compressed sounds were encoded and decoded using the LAME MP3 encoder [25]. Instrument sounds were compressed with three different bit rates (32, 56, and 112 Kbps). These three bit rates gave near-perfect (for 32 Kbps), intermediate (for 56 Kbps), and near-random discrimination (for 112 Kbps) in a previous discrimination study of these MP3-compressed musical instrument sounds [16].

The subjects compared the stimuli in terms of ten emotional categories: Happy, Heroic, Romantic, Comic, Calm, Mysterious, Shy, Angry, Scary, and Sad. We carefully picked the emotional categories based on terms we felt composers were likely to write as expression marks for performers (e.g., *mysteriously*, *shyly*, etc.) and at the same time would be readily understood by lay people. The subjects were provided with an instruction sheet containing definitions of the ten emotional categories from the Cambridge Academic Content Dictionary [26]. Every subject made paired comparisons between the sounds.

The test asked listeners to compare four types of compressed sounds for each instrument over ten emotion categories. During each trial, subjects heard a pair of sounds from the same instrument with different types of compression (no compression, 112Kbps, 54Kbps, and 32Kbps) and were prompted to choose which sounded stronger for given emotional characteristics. This method was chosen for simplicity of comparison, since subjects only needed to remember two sounds for each comparison and make a binary decision. This required minimal memory from the subjects, and allowed them to give more instantaneous responses [19, 4, 27].

Each combination of two different compressions was presented for each instrument and emotion category, and the listening test totaled $P_2^4 \times 8 \times 10 = 960$ trials. For each instrument, the overall trial presentation order was randomized (i.e., all combinations of compressed bassoon sounds were in a random order, then all the clarinet comparisons, etc.). However, the emotional categories were presented in order to avoid confusing and fatiguing the subjects. The listening test took about 2 hours, with a short break of 5 minutes after every 30 minutes to help minimize listener fatigue and maintain consistency.

4. RANKING RESULTS FOR THE EMOTIONAL CHARACTERISTICS WITH DIFFERENT OF MP3 BIT RATES

We ranked the compressed sounds by the number of positive votes they received for each instrument and emotion, and derived scale values using the Bradley-Terry-Luce (BTL) statistical model [28, 29]. For each instrument-emotion pair, the BTL scale values for the original and three compressed sounds sum to 1. The BTL value for each sound is the probability that listeners will choose that compression rate when considering a certain instrument and emotion category. For example, if all four sounds (the original and three compressed sounds) are judged equally happy, the BTL scale values would be 1/4=0.25. We also derived the corresponding 95% confidence intervals for the compressed sounds using the method proposed by Bradley [28].

Fig. 1 to 6 show the BTL values and corresponding 95% confidence intervals for each emotional category. Based on the data in Fig. 1 - 6, Table 1 shows the number of instruments that were significantly different from the original sound (i.e., the 95% confidence intervals of the original and compressed sounds did not overlap) for each compression rate and emotional category. The table shows that there were relatively few differences for 112 and 56Kbps, but most of the instruments were significantly different for 32Kbps in nearly every category. This agrees with the results of Lee et al. [16], which found very good discrimination between the original and compressed sounds at 32Kbps, but poor discrimination at 56 and 112Kbps.

To help understand which instruments and emotional categories were most and least affected by MP3 compression, Table 2 shows the number of compressed sounds that were significantly different from the original sound for each instrument and emotional category. Based on the data, the clarinet was the most affected instrument (closely followed by the oboe and saxophone), while the horn was by far the least affected instrument. Lee et al. [16] also found the MP3-compressed horn relatively more difficult to discriminate from the original compared to other instruments. Among emotional categories in Table 2, Happy and Calm were the most affected, and Angry was by far the least affected.

Fig. 7 shows how often the original instruments sounds were statistically significantly greater than the three compressed sounds (This is different than the sum in the final column of Table 2 which counts any significant difference - both those significantly greater and those significantly less). Positive values indicate an increase in the emotional characteristics, and negative values a decrease. Again, Happy and Calm were the most affected emotional characteristics. Emotional categories with larger Valence (e.g., Happy, Heroic, Romantic, Comic, Calm) tended to decrease with more MP3 compression, while emotional categories with smaller Valence (e.g., Sad, Scary, Shy, Mysterious) tended to increase with more MP3 compression. As an exception, Angry was relatively unaffected by MP3 compression for the compression rates we tested.

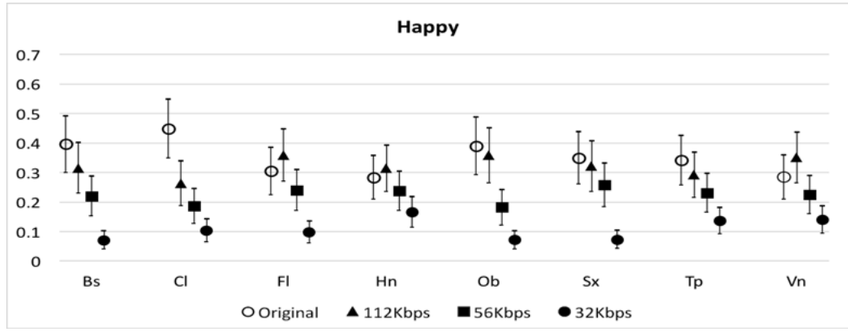


Figure 1. BTL scale values and the corresponding 95% confidence intervals for the emotional category Happy.

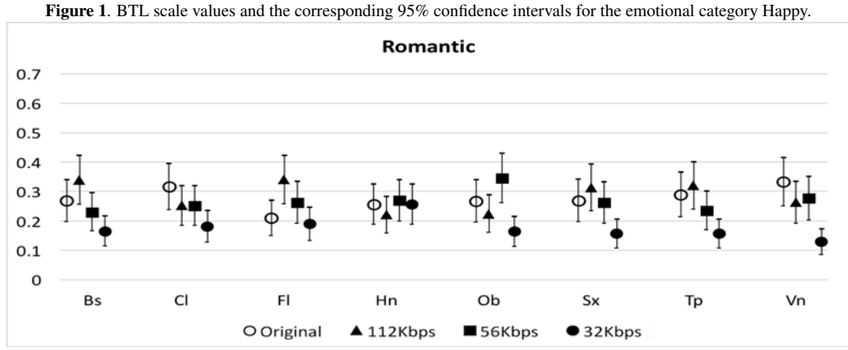


Figure 2. BTL scale values and the corresponding 95% confidence intervals for Romantic.

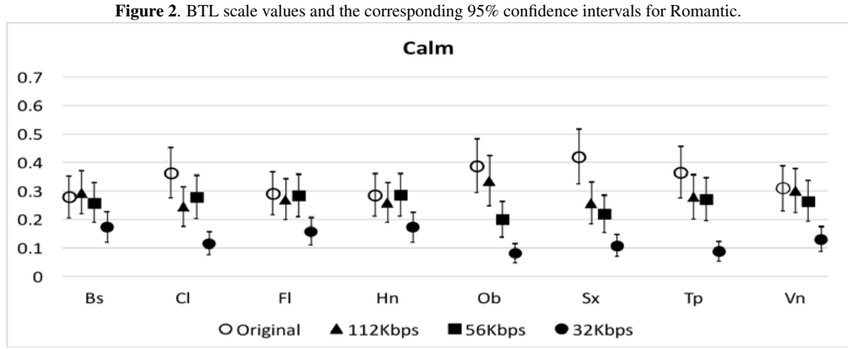


Figure 3. BTL scale values and the corresponding 95% confidence intervals for Calm.

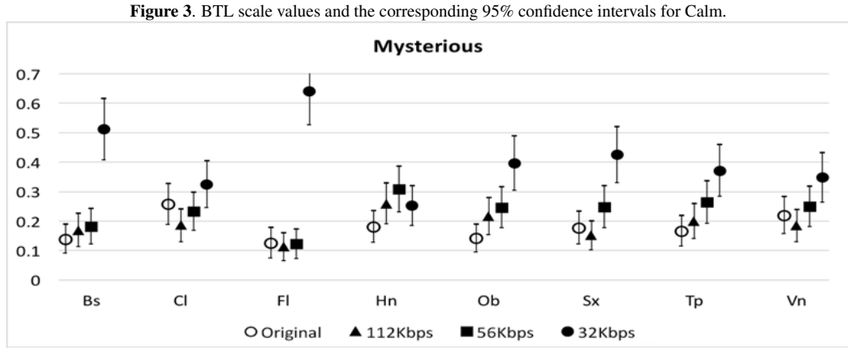


Figure 4. BTL scale values and the corresponding 95% confidence intervals for Mysterious.

5. DISCUSSION

The goal of our work was to understand how emotional characteristics of instruments vary with MP3 compression. Based on the Table 2 and Figure 7, our main findings are as follows:

1. Negative and neutral emotional characteristics (Sad, Scary, Shy, and Mysterious) increased with more MP3 compression in the samples we tested (see Figure 7).
2. Positive emotional characteristics (Happy, Heroic, Romantic, Comic, and Calm) decreased with more MP3 compression in the samples we tested (see Figure 7).
3. Angry was relatively unaffected by MP3 compression for the rates we tested (see Figure 7).
4. MP3 compression affected some instruments more and others less. The clarinet, oboe, and saxophone were most affected, and the horn by far the least affected (see Table 2).

As a possible explanation for these results, perhaps quantization jitter introduced into the amplitude envelopes by MP3 compression decreased positive emotional characteristics such as Happy and Calm while increasing others such as Mysterious by changing the quality of sounds to be somewhat different and unnatural. The above results demonstrate how a categorical emotional model can give more emotional nuance and detail than a 2D dimensional model with only Valence and Arousal. For example, Scary and Angry are very close to each one another in terms of Valence and Arousal, yet Scary was significantly increased with more compression while Angry was relatively unaffected. The results suggest that they are distinctively different emotional characteristics.

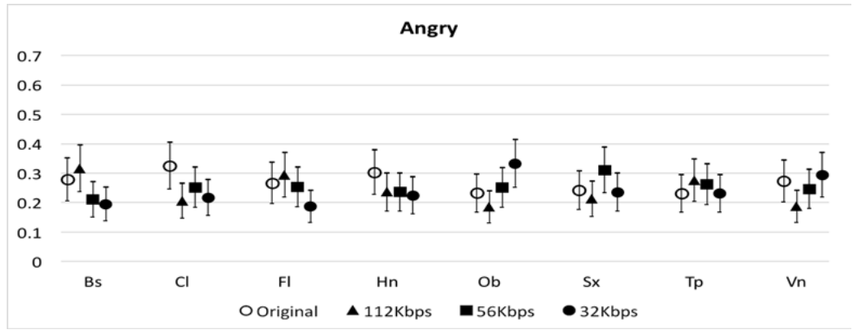


Figure 5. BTL scale values and the corresponding 95% confidence intervals for Angry.

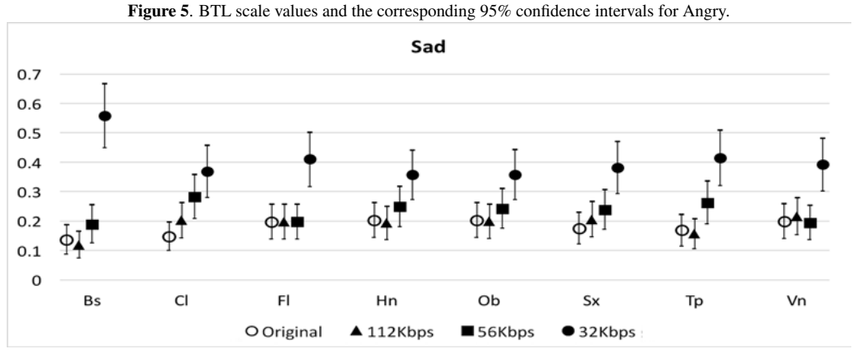


Figure 6. BTL scale values and the corresponding 95% confidence intervals for Sad.

Emotional Category	112Kbps	56Kbps	32Kbps
Happy	1	3	8
Heroic	0	1	7
Romantic	1	0	6
Comic	0	2	5
Calm	2	2	8
Mysterious	0	2	6
Shy	1	0	8
Angry	1	0	1
Scary	0	2	7
Sad	0	1	8
Avg.	0.6	1.3	6.4

Table 1. The number of instruments that were significantly different from the original sound (i.e., the 95% confidence intervals of the original and compressed sounds did not overlap) for each compression rate and emotional category

Emotional Category	Bs	Cl	Fl	Hn	Ob	Sx	Tp	Vn	Total
Happy	2	3	1	1	2	1	1	1	12
Heroic	1	1	2	0	1	1	1	1	8
Romantic	1	1	1	0	1	1	1	1	7
Comic	1	1	1	1	1	1	1	1	8
Calm	1	2	1	1	2	3	1	1	12
Mysterious	1	0	1	0	2	1	1	1	7
Shy	1	1	1	1	1	2	1	1	9
Angry	0	1	0	0	0	0	0	0	1
Scary	1	1	1	0	1	1	2	1	8
Sad	1	2	1	1	1	1	1	1	9
Total	10	13	10	5	12	12	10	9	

Table 2. The number of compressed sounds that were significantly different from the original sound (i.e., the 95% confidence intervals of the original and compressed sounds did not overlap) for each instrument and emotional category.

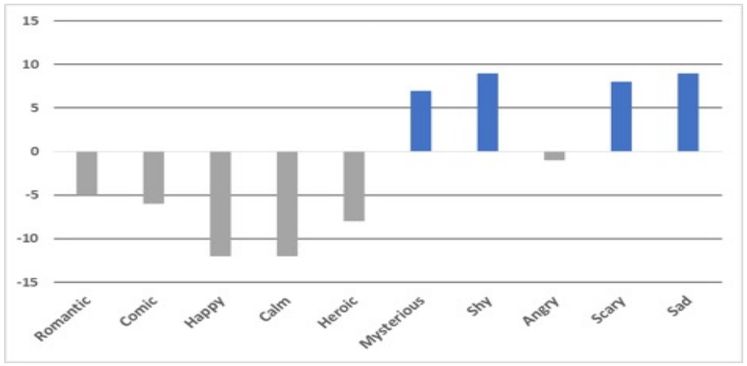


Figure 7. The number of significant differences between the original and compressed sounds, where strengthened emotional categories are positive, and weakened emotional categories are negative.

6. REFERENCES

[1] B. Sisario, "Downloads in Decline as Streamed Music Soars," *The New York Times (New York edition)*, p. B3, July 2014.

[2] T. Eerola, R. Ferrer, and V. Alluri, "Timbre and Affect Dimensions: Evidence from Affect and Similarity Ratings and Acoustic Correlates of Isolated Instrument Sounds," *Music Perception: An Interdisciplinary Journal*, vol. 30, no. 1, pp. 49–70, 2012.

[3] B. Wu, A. Horner, and C. Lee, "Musical Timbre and Emotion: The Identification of Salient Timbral Features in Sustained Musical Instrument Tones Equalized in Attack Time and Spectral Centroid," in *International Computer Music Conference (ICMC), Athens, Greece*, 14-20 Sept 2014, pp. 928–934.

[4] C.-j. Chau, B. Wu, and A. Horner, "Timbre Features and Music Emotion in Plucked String, Mallet Percussion, and Keyboard Tones," in *International Computer Music Conference (ICMC), Athens, Greece*, 14-20 Sept 2014, pp. 982–989.

[5] B. Wu, C. Lee, and A. Horner, "The Correspondence of Music Emotion and Timbre in Sustained Musical Instrument Tones," *Journal of the Audio Engineering Society*, vol. 62, no. 10, pp. 663–675, 2014.

[6] C.-j. Chau, B. Wu, and A. Horner, "The Emotional Characteristics and Timbre of Nonsustaining Instru-

ment Sounds," *Journal of the Audio Engineering Society*, vol. 63, no. 4, pp. 228–244, 2015.

[7] T. Zwicker, "Psychoacoustics as the basis for modern audio signal data compression," *The Journal of the Acoustical Society of America*, vol. 107, no. 5, pp. 2875–2875, 2000.

[8] S. van de Par and A. Kohlrausch, "Three approaches to the perceptual evaluation of audio compression methods," *The Journal of the Acoustical Society of America*, vol. 107, no. 5, pp. 2875–2875, 2000.

[9] M. Erne, "Perceptual Audio Coders" What to listen for"," in *Audio Engineering Society Convention 111*. Audio Engineering Society, 2001.

[10] C.-M. Chang, H.-W. Hsu, K.-C. Lee, W.-C. Lee, C.-M. Liu, S.-H. Tang, C.-H. Yang, and Y.-C. Yang, "Compression artifacts in perceptual audio coding," in *Audio Engineering Society Convention 121*. Audio Engineering Society, 2006.

[11] P. Marins, "Characterizing the Perceptual Effects Introduced by Low Bit Rate Spatial Audio Codecs," in *Audio Engineering Society Convention 131*. Audio Engineering Society, 2011.

[12] H. Fuchs, W. Hoeg, and D. Meares, "ISO/MPEG subjective tests on multichannel audio systems: design and methodology," in *Broadcasting Convention, 1994. IBC 1994., International*, Sep 1994, pp. 152–157.

[13] D. Kirby, F. Feige, and U. Wustenhagen, "ISO/MPEG subjective tests on multichannel audio coding systems: practical realisation and test results," in *Broadcasting Convention, 1994. IBC 1994., International*, Sep 1994, pp. 132–139.

[14] W. Schmidt and E. Steffen, "ISO/MPEG subjective tests on multichannel audio coding systems: statistical analysis," 1994.

[15] G. Stoll and F. Kozamernik, "EBU subjective listening tests on low-bitrate audio codecs," 2003.

[16] C. Lee and A. Horner, "Discrimination of MP3-Compressed Musical Instrument Tones," *Journal of the Audio Engineering Society*, vol. 58, no. 6, pp. 487–497, 2010.

[17] C. Lee, A. Horner, and B. Wu, "The Effect of MP3 Compression on the Timbre Space of Sustained Musical Instrument Tones," *Journal of the Audio Engineering Society*, vol. 61, no. 11, pp. 840–849, 2013.

[18] S. McAdams, J. W. Beauchamp, and S. Meneguzzi, "Discrimination of musical instrument sounds resynthesized with simplified spectrotemporal parameters," *The Journal of the Acoustical Society of America*, vol. 105, no. 2, pp. 882–897, 1999.

[19] A. Horner, J. Beauchamp, and R. So, "Detection of random alterations to time-varying musical instrument spectra," *The Journal of the Acoustical Society of America*, vol. 116, no. 3, pp. 1800–1810, 2004.

[20] J. W. Beauchamp, A. B. Horner, H.-F. Koehn, and M. Bay, "Multidimensional scaling analysis of centroid-and attack/decay-normalized musical instrument sounds," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 3276–3276, 2006.

[21] A. B. Horner, J. W. Beauchamp, and R. H. So, "Detection of time-varying harmonic amplitude alterations due to spectral interpolations between musical instrument tones," *The Journal of the Acoustical Society of America*, vol. 125, no. 1, pp. 492–502, 2009.

[22] ——, "Evaluation of mel-band and mfcc-based error metrics for correspondence to discrimination of spectrally altered musical instrument sounds," *Journal of the Audio Engineering Society*, vol. 59, no. 5, pp. 290–303, 2011.

[23] M. Bosi and R. E. Goldberg, *Introduction to digital audio coding and standards*. Springer Science & Business Media, 2012, vol. 721.

[24] C. Lee, A. Horner, and J. Beauchamp, "Discrimination of Musical Instrument Tones Resynthesized with Piecewise-Linear Approximated Harmonic Amplitude Envelopes," *J. Audio Eng. Soc*, vol. 60, no. 11, pp. 899–912, 2012.

[25] LAME MP3 Encoder. [Online]. Available: http://lame.sourceforge.net/

[26] Cambridge Academic Content Dictionary. [Online]. Available: http://dictionary.cambridge.org/dictionary/american-english

[27] R. Mo, B. Wu, and A. Horner, "The Effects of Reverberation on the Emotional Characteristics of Musical Instruments," *J. Audio Eng. Soc*, vol. 63, no. 12, pp. 966–979, 2016. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=18055

[28] R. A. Bradley, "Paired comparisons: Some basic procedures and examples," *Nonparametric Methods*, vol. 4, pp. 299–326, 1984.

[29] F. Wickelmaier and C. Schmid, "A Matlab Function to Estimate Choice Model Parameters from Paired-comparison Data," *Behavior Research Methods, Instruments, and Computers*, vol. 36, no. 1, pp. 29–40, 2004.

COMPOSITION as an EVOLVING ENTITY
an EXPERIMENT in PROGRESS

Sever Tipei
Computer Music Project
University of Illinois
s-tipei@illinois.edu

ABSTRACT

Composition as an Evolving Entity envisions a work in continuous transformation, never reaching equilibrium, a complex dynamic system whose components permanently fluctuate and adjust to global changes. The process never produces a definitive version, but at any arbitrary point in time provides a plausible variant of the work - a transitory being. Directed Graphs are used to represent the structural levels of any composition (vertices) and the relationships between them (edges). By determining adjacencies and degrees of vertices and introducing weights for edges, one can define affinities and dependencies. Ways in which the all-incidence matrix of a graph with weighted edges can evolve are discussed including the use of Information Theory. The Evolving Composition model is closer to the way composers actually write music and refine their output; it also creates the equivalent of a live organism, growing, developing, and transforming itself over time.

1. BACKGROUND

The process of writing a new piece involves balancing elements of different structural levels from the overall form of the composition to various sound characteristics. In the article "Morphogenetic Music" [1], composer Aurel Stroe and his collaborators discussed the play between melody, rhythm, harmony, and phrase length in Mozart's *Piano Sonata in C Major K.W. 309* showing how unexpected or daring choices at one structural level are compensated by blander, more familiar occurrences at others. A related insight is given by Beethoven's sketchbooks that show a constant adjustment, sometimes over years, of initial motives [2] and in the works of Charles Ives who kept modifying his music after it was published.

These universal concerns also apply to contemporary works and are shared regardless of aesthetics, historical moment or style. In electro-acoustic music, readily available software allows authors to investigate alternatives in placing gestures, textures, structural elements, etc. or to further adapt and polish the sound materials after the completion of the project.

1.1 Manifold Compositions

When a computer-generated piece contains elements of indeterminacy, multiple variants can be produced by changing the initial conditions (eg. the random number generator's seed). Randomness may be involved in selecting the order of macro and micro events, in the choice of attack times and durations of sounds, of their frequencies, amplitudes, spectra, etc. or of their environment's properties such as location in space and reverberation. Such multiple variants, members of a *manifold composition*, have the same structure and are the result of the same process but differ in the way individual events are distributed in time: like faces in a crowd, they all share basic features but exhibit particular attributes. A *manifold composition* is an equivalence class, a composition class, produced by a computer under particular conditions [3]. It includes all its actual and virtual variants and requires that all of them be equally acceptable. *Manifold compositions* follow the example of Stockhausen (*Plus-minus*) [4], Xenakis (*ST* pieces) [5] and Koenig (*Segmente*) [6] and extend it: by stipulating the use of a computer and introducing elements of indeterminacy during the act of composing in the case of Stockhausen; by adding more constraints in the case of Xenakis and Koenig.

1.2 DISSCO

The software used in the production of *manifolds*, DISSCO provides a seamless approach to composition and sound design [7]. An integrated environment, it has three major parts: LASS, a Library for Additive Sound Synthesis, which builds sounds from first principles, CMOD, or Composition MODule, a collection of methods for composition driving the synthesis engine, and LASSIE, a graphic user interface (GUI).

DISSCO is comprehensive in the sense that it does not require the intervention of the user once it starts running. This kind of "black box" set of instructions is necessary for preserving the integrity of *manifold* production: modifying the output or intervening during computations would amount to the alteration of data or of the logic embedded in the software. Due to an option unavailable on other systems, the control of the *perceived loudness*, a non-linear function of amplitude [8], post-production interventions become not only unnecessary but also incongruent with the purpose of the enterprise.

1.3 Indeterminacy

In DISSCO, randomness is introduced through uniform (flat) distributions by the RANDOM method or made available through envelopes (functions). A library, ENVLIB allows the composer to draw the contour of the

curve, scale and store it while MAKE ENVELOPE offers the possibility to enter a list of ***x*** and ***y*** values and to specify a range within which each of them may randomly fluctuate. Stochastic distributions expressions are handled with the help of the *muparser* [9].

Two other options are introduced by STOCHOS: 1) a dynamic range whose min. and Max. limits are defined by two envelopes while a third one controls the distribution within the confined area and 2) multiple probability ranges whose sum is 1 at any moment (inspired by Xenakis' special density diagram determining orchestration) [10]. Finally, VALUEPICK, introduces weighted probabilities assigned to discrete values at any parameter.

2. DIRECTED GRAPHS

The structure of CMOD can be represented as a directed graph (DG), a *rooted tree,* where every level inherits from a generic Event class in a *matryoshka* type of arrangement: a unique Top event (the root) can include High followed by Mid, Low, and Bottom events - the platform where individual sounds are created. In this model, events are represented as vertices each of them having siblings (except the root) and spawning any number of children. They are connected by edges that illustrate the relationships between them. By carefully determining adjacencies and degrees of all vertices and by introducing weights for edges, one can start defining affinities and dependencies in a musical composition. The scheme can accommodate both the stricter order found in traditional music (piece < sections < themes < motives < cells < sounds), and, random distribution of undifferentiated events (sounds in Cage's chance music works) if only the root and its children are present. Moreover, this model is well suited to the creation of "floating hierarchies", unstable flows of information that favor change over established formulations [11].

2.1 Similar Approaches

Pierre Barbaud had explored the use of graphs in "automatizing" the production of tonal harmonic and contrapuntal sequences in his own works as early as the 1960s [12] and there is a similarity between DGs and the arborescences on which many later works of Xenakis are predicated. More recently, a number of authors have either proposed formalisms and/or built musical systems based on Directed Graphs. Among them, *Nodal* a system for generative composition [13] and *Graph Theory*, a piece by Jason Freeman [14] are the closest to the tenor of this project.

Evolving Composition project adopts the point of view that one way in which any musical composition can be described is as a rooted tree DG. It is a framework that corresponds *post factum* to the way CMOD is organized, and it is informed by musical practice. Not intended as a way to generate pitches, rhythms, etc. or to explore the limits of creativity like other schemes, it is used to represent relations between structural components of a musical work - a goal that could be expanded in the future.

Relevant to possible future developments is Jonathan Owen Clark's formalism [15] that brings together Graphs

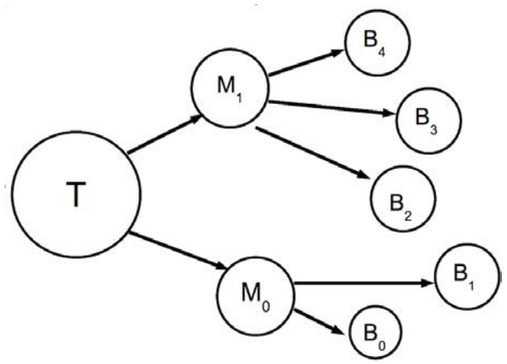


Figure 1. DISSCO structure as a rooted directed graph. For clarity only one intermediate level (M) is shown.

and Dynamic Systems. It could be applied to the content of terminal vertices - sounds in a multidimensional vector space - and to their influence on the macro levels.

3. COMPLEX DYNAMIC SYSTEMS

Any composition can be thought of as a complex system. During the process of composing it, the system is also dynamic since options are constantly re-evaluated. This leads to changes both in the macro structure and in the details of the work that are not necessarily chaotic.

The Evolving Composition project models such a process by allowing the computations to continue for an arbitrary amount of time. It envisions a work in perpetual transformation, never reaching an equilibrium, a complex structure whose components permanently fluctuate and adjust to each other's modifications - a "brewing" piece. Such a composition can be regarded as a network of evolving interdependent elements whose alterations result in a series of unstable dynamic states. It could be likened to an electric grid where power is generated and distributed through different nodes: the grid has to be responsive and to constantly adjust the flow of electricity to compensate for surges in demand or for local failures. Its musical equivalent is a composition whose parts are interconnected at all levels in such a way that modifying one component could have global consequences and affect other parts of the system.

This view of the composition as a network of perpetually unfolding elements in search of an elusive balance, similar to a living creature, epitomizes an "organic" approach to creating music. The process never produces a definitive version but provides at any arbitrary point in time a plausible variant of the work - a transitory being.

Composition as an Evolving Entity is an augmentation and a corollary of the *manifold* idea as they both generate an unlimited number of variants, involve the presence of randomness at all structural levels, and relay on the view of sounds as events in a multidimensional vector space whose degrees of freedom include time/duration, frequency, amplitude, etc. The project is predicated on discovering and creating new situations as opposed to attaining known, already established goals: a volatile equilibrium and NOT a search for a stable optimal solution.

4. THE DESIGN

4.1 Trivial Case

Upon finishing a new piece, a human composer might step back, take a fresh look at the work and, possibly, decide on making changes and adjustments. The Evolving Entity allows computations to continue after the first variant of the *manifold* is completed: a new edge is created between the last Bottom event $\mathbf{X}_{last}$, (a terminal vertex) and another vertex $\mathbf{X}_{new}$ which could be a sibling, a parent or an ancestor belonging to the same branch or to a different one. The operation takes place with the help of an all-incidence matrix $\mathcal{M}$ of the type shown in Figure 2.

This transitional matrix is weighted (probabilities assigned to different edges) and serves as a template for the Evolving Entity, a sort of genome of the composition.

	T	M_0	M_1	B_0	B_1	B_2	B_3	B_4
T	0.01	0.05	0.05	0.02	0.01	0.03	0.03	0.01
M_0	0.20	0.01	0.25	0.20	0.20	0.05	0.10	0.07
M_1	0.20	0.01	0.01	0.11	0.10	0.25	0.20	0.23
B_0	0.12	0.20	0.11	0.01	0.30	0.19	0.08	0.07
B_1	0.12	0.24	0.10	0.35	0.01	0.07	0.08	0.09
B_2	0.12	0.06	0.15	0.10	0.12	0.01	0.26	0.26
B_3	0.12	0.08	0.18	0.11	0.14	0.25	0.01	0.26
B_4	0.11	0.06	0.15	0.11	0.12	0.25	0.24	0.01

Figure 2. All-incidence weighted matrix

The selection of $\mathbf{X}_{new}$ involves dividing the components of the vector $\mathbf{V}_{last}$ (corresponding to $\mathbf{X}_{last}$) by their sum, adding the results in order and matching a random number to one of the probability intervals thus created. If the newly chosen vertex $\mathbf{X}_{new}$ is a parent, all its descendents are computed anew. Upon completion an audio file becomes available to be examined and a vector $\mathbf{V}_{new}$ corresponding to the chosen vertex $\mathbf{X}_{new}$ is used to continue. The procedure may be repeated an arbitrary number of times.

4.2 Continuity

When the process of re-evaluating vertices proceeds without interruption, the sequence of pseudorandom numbers creates a history uniquely determined by the seed and confers the Evolving Composition Entity the equivalent of a "personal" identity. There is a paradox here: the choices leading to any variant of the manifold depend on chance but the random numbers themselves are part of a causal chain. Since the directed graph and the matrix/genome are pre-determined, a balance is created between structure, and indeterminacy, and the Evolving Piece starts to resemble a living organism whose cells are rejuvenated constantly while the creature endures.

4.3 Template Modification

Modifications of the template/genome could be introduced as computations continue. If the column vector $\mathbf{V}_{last}$ is multiplied by the matrix $\mathcal{M}$, $\mathbf{V}_{last} * \mathcal{M}$, every time a new variant of the piece completes, a Markov chain mechanism is initiated and the newly resulting vector $\mathbf{V}_{last+1}$ becomes part of an ordered sequence of causally connected vectors.

The user controls the likelihood of various connections/edges between vertices through the static, all-incidence matrix $\mathcal{M}$. The Markov chain mechanism allows a vector to evolve in a predictable way but assumes that the content of the other vector/columns of the matrix remain the same. A more realistic alternative is to take into account global changes that might occur every time a new version is computed - something a human composer would probably do.

Such adjustments are construed as the result of the composer's intuition, taste, training, etc. but many times these subjective considerations can also be described using elements of Information Theory. The main concepts provided by Information Theory as applied to musical messages are those of Entropy/Order - expressed through the relationship between Originality and Redundancy - in relation to the Complexity of the work [16]. Their relevance to this project is based on at least two facts: these are measurable quantities and, as Herbert Brün once put it: "the job of a composer is to delay the decay of information".

As an example, Originality may be equated with i*m*probability hence with the delivered Information, Redundancy with repetition and/or familiarity, and Complexity with the number of available choices, all quantifiable if not entirely in an objective way. Since each variant of the piece exhibits new, different values for most vertices, an analysis of all values at all vertices followed by a comparison with a desired situation becomes necessary. In turn, such an extensive re-evaluation of data requires a significant increase in computing time and storage capacity since even a relatively short work may easily contain hundreds of vertices.

Moreover, the vertices representing the Bottom level contain significantly more information then those corresponding to higher levels and are likely to trigger more often global changes. This is because sound design procedures are concentrated at the Bottom level: various ways of assigning the frequency and loudness of a sound, the rate and amplitude of vibrato (FM), of tremolo (AM) or of frequency and amplitude transients. Information about spatialization and reverberation should also be added to the list.

4.4 Developing Entity

The Complex Dynamic System that is the Evolving Composition includes the DG rooted tree DISSCO, the template/genome matrix $\mathcal{M}$, and the set of data used to create the initial variant of the piece. So far we assumed constant the size of the rooted tree and that of the matrix. However, the process could start with a tree and a matrix reduced to a small number of vertices/vectors, for instance only the Top vertex (the piece) and one or two terminal vertices. The system could then be allowed to grow

by developing more edges and vertices until reaching its maximum potential. The opposite, a decaying slope, can be engineered by cutting off branches of the tree and gradually reducing the matrix size. In the end, a minimum number of vertices or the Markov chain reaching the ergodic state could trigger the demise of the Entity. Using another analogy, the growing number of vertices and edges in the first stage of its evolution mirrored by a reduction of the network toward the end could be associated with the growing number of neurons and synapses during the human infancy and the pruning that occurs during adolescence.

5. IMPLEMENTATION

5.1 Why DISSCO ?

Evolving Composition uses the structure and features of DISSCO, a powerful software that has been proven reliable and robust during an almost a decade of use both by seasoned users and students. DISSCO offers an unbroken link between a Computer-assisted Composition module that offers deterministic tools (patterns, sieves, etc.) along with random distributions and a synthesis engine with uncommon capabilities (eg. control of perceived loudness) that generates - according to users - a sound output superior to many other similar applications. The project being an extension of the *manifold* undertaking, DISSCO was the obvious choice.

5.2 Present phase

The general framework described above was selected after considering a number of alternatives. The Trivial Case was implemented by connecting the last Bottom event to the Top event without interrupting the sequence of random numbers. This first stage of the project is now used to develop *Sound Fountai*n, an installation producing continuous sound output in a local building's atrium.

Presently, the Evolving Composition project runs in multithreading mode and has been recently ported on a multi-core system. Using 16 CPU cores when realizing a complex eight channel piece, the ratio between computation time and duration of the piece (real time) is a little less than 3/2; increasing the number of cores does not result in a significant improvement. In other examples, a six minute stereo piece ran on the same system in less than five minutes while an experiment in granular synthesis, of a few minutes and over 350,000 grains with a dozen partials each took over three hours.

Computing time depends heavily on both the complexity of sounds, and their duration. DISSCO was conceived as a “Rolls Royce bulldozer” (refined control over large numbers of elements) running on high-performance computers: it allows for an arbitrary number of partials and envelope segments along with involved ways of selecting sound attributes. However, a meaningful functioning of the system requires a ratio of 1/1 or better and a urgent task is to profile, optimize, and parallelize the code in order to constantly achieve real time or faster.

5.3 Future work

This is an experiment in progress in its incipient stage. and some aspects still need to be worked out. Conceptually, the project is situated at the intersection of Dynamic Systems Theory, Graph Theory, Information Theory, and high-performance computing. A solid and practical link between them still needs to be formulated.

From a computational point of view, efficient ways of creating the $\mathcal{M}$ matrix need to be explored. As an example, a recent work contains 132 discrete event types (vertices): a 132 X 132 matrix or matrices an order of magnitude higher are manageable but they will have to be constantly updated and various operations performed on them. In case elements of Information Theory are used, an evaluation of each new variant of the piece is necessary meaning information for 15,700 sounds (as in the above example) or more will have to be not only stored but also analyzed.

6. CONCLUSIONS

Complex Dynamic Systems and Graph Theory have been discussed in relation to Catastrophe Theory and Morphogenetic Music by both Stroe [1] and Clark [15], in the context of (pseudo-)tonal music [17], or by considering music a language but, to our knowledge, no mechanism generating an evolving composition as a result of uninterrupted computations has been proposed..

The Emerging Entity composition model is closer to how humans actually compose, by trial and error, continuously refining the output. It also reflects the natural world by creating (like some Artificial Life projects) the equivalent of a live organism, growing, developing, transforming itself over time and thus fulfilling the goal expressed by John Cage: “ to imitate nature in its mode of operation”.

This paradigm can be developed beyond the immediate scope of this proposal by creating an “ecosystem” where the performance environment (hall acoustics) and live performers' decisions influence the composition.

Acknowledgments

We would like to acknowledge the support of Dr. Volodymyr Kindratenko and the Innovative Systems Laboratory at National Center for Supercomputing Applications (NCSA), for facilitating the computing infrastructure to perform the work.

REFERENCES

[1] A. Stroe, C.Georgescu, and M.Georgescu, ”Morphogenetic Music”, unpublished manuscript, Bucharest, cca. 1985.

[2] W. Kinderman, *Artaria 195, Beethoven's Sketchbook for the Missa solemnis and the Piano Sonata in E Major, Opus 109*, University of Illinois Press, Urbana, 2003.

[3] S. Tipei, “Manifold Compositions – a (Super)computer-assisted Composition Experiment in Progress”, *Proc. 1989 Int'l Computer Music Conference*, Ohio State University, Columbus, OH, 1989, pp. 324-327.

[4] K. Stockhausen, *Plus-minus*, Universal Edition, London, 1965

[5] I. Xenakis , *ST* pieces, Boosey and Hawkes, London, NewYork, 1967.

[6] M. G. Keonig, *Segmente*, Tonos Musikverlags GmbH, Darmstadt, Germany, 1983.

[7] H. G. Kaper and S. Tipei, “DISSCO: a Unified Approach to Sound Synthesis and Composition”, Proc. *2005 Int'l Computer Music Conference,* Barcelona, Spain, September 2005, pp. 375-378.

[8] J. Guessford, H. G. Kaper, and S.Tipei. “Loudness Scaling in a Digital Synthesis Library”, Proc. *2004 Int'l Computer Music Conference, Miami*, Florida, November 2004, pp. 398-401.

[9] - *muparser - Fast Math Parser Library*, http://beltoforion.de/article.php?a=muparser&p=features.

[10] I. Xenakis, *Formalized Music*, Pendragon Press, Stuyvesant, NY, 1992, p. 139.

[11] H. Brün, “On Floating Hierarchies”, talk given at *American Society for Cybernetics*, Evergreen College, October 20,1982,http://ada.evergreen.edu/~arunc/texts/brun/pdf/brunFH.pdf, accessed,September 7, 2015.

[12] P. Barbaud, *Initiation a la composition musicale automatique,* Dunod, Paris, 1966.

[13] J. McCormack, P. McIlwain, A. Lane and A. Dobrin - “Generative Composition with Nodal”, *Workshop on Music and Artificial Life*, Lisbon, Portugal, 2007.

[14] J. Freeman, Graph Theory, http://archive.turbulence.org/Works/graphtheory/, accessed April 23, 2016

[15] J. O. Clark, “Nonlinear Dynamics of Networks: Applications to Mathematical Music Theory”, in *Mathematics and Computation in Music*, Springer, Berlin, 2009, pp. 330-339.

[16] A. Moles, *Information Theory and Aesthetic Perception*, University of Illinois Press, Urbana, 1958.

[17] E. W. Large, “A Dynamical Systems Approach to Music Tonality”, *www.ccs.fau.edu/~large/Publications/Large2010Tonality.pdf*, accessed August 19, 2015.

Nodewebba: Software for Composing with Networked Iterated Maps

Bret Battey
Music, Technology and Innovation Research Centre
De Montfort University
bbattey@dmu.ac.uk

ABSTRACT

Nodewebba *software provides a GUI implementation of the author's "Variable-Coupled Map Networks" (VCMN) approach to algorithmic composition. A VCMN node consists of a simple iterated map, timing controls, and controls for mapping the node output to musical parameters. These nodes can be networked, routing the outputs of nodes to control the variables of other nodes. This can enable complex emergent patterning and provides a powerful tool for creating musical materials that exhibit interrelated parts.* Nodewebba *also provides API hooks for programmers to expand its functionality. The author discusses the design and features of* Nodewebba*, some of the technical implementation issues, and a brief example of its application to a compositional project.*

1. INTRODUCTION

The use of pseudo-random number generators is foundational to many classical algorithmic music techniques. One well-established approach for generating pseudo-random numbers is to use Lehmer's Linear Congruence formula (LLCF) [1], an iterative map:

$$x_t = (x_{t-1}a + b) \bmod m \qquad (1)$$

The variables are optimized to provide a maximal possible period of non-repetition for a given computer's architecture. LLC falls short of uniform randomness in the short term [1], and in fact tuples of successive values exhibit a type of lattice structure [2]. Of course, pure uniformity isn't a necessity for most algorithmic-music applications. In fact, if one significantly deoptimizes its variables, LLCF can become a useful and flexible pattern generator, exhibiting a range of behaviors from decay or rise to steady state, simple periodicity, layered periodicity or unpredictable self-similarity. In a previous paper [3], I analyzed this range of behaviors and provided some guidelines to working with its parameter space. Later work by Warren Burt demonstrated an approach to LLCF that utilized a broader range of *a* and *b* variables to enable compositional exploration of varying degrees of "almost random" behavior [4]. Recent audiovisual artwork by Francesc Pérez utilizes LLCFs to control video and audio granulation processes [5,6].

The primary purpose of my aforementioned paper was to introduce the concept of Variable-Coupled Map Networks (VCMNs). A VCMN consists of a set of interlinked nodes. The core of each node is an iterated map function. The output of any one of these nodes may set a function variable in itself or any of the other nodes. Each node has a wait-time between iterations, which can also be set or controlled by other nodes. The node outputs can then be mapped to musical parameters. In theory, a node can be any iterative map, but my research focused solely on LLCF, due to its simplicity of implementation, fixed output range and small number of variables.

VCMN configurations, particularly those including feedback mechanisms, can readily exhibit "emergence", where "properties at a certain level of organization… cannot be predicted from the properties found at the lower levels." [7] Further, the paper demonstrated counterpoint behaviors between the nodes could arise when those nodes were mapped to multiple note streams or note parameters. Also notable was the capacity of VCMNs to make coherent rhythmic gestures even with entirely non-quantized timing values.

VCMNs had the disadvantage of still requiring coding to implement, making it time-consuming to configure and alter networks and explore their musical potentials. Nonetheless, I used the approach to generate some crucial materials in several compositions in the 1990s and 2000s. It was only with my 2011 audiovisual work *Clonal Colonies*[1] for the Avian Orchestra that I used VCMN as a primary tool in the creation of a large-scale work. The code I developed for and insights derived from composing the work inspired the creation of *Nodewebba*. An open-source project developed in the Max 7[2] programming language, it provides a GUI-based environment for using LLC and VCMN for pattern generation for music and other media. It provides both MIDI and floating-point outputs for mapping to a variety of targets, and it readily allows programmers to add additional functionality and integrate *Nodewebba* with other projects.

[1] Available at http://BatHatMedia.com/Gallery/clonal.html
[2] https://cycling74.com

2. FEATURES AND INTERFACE

2.1 Node Interface

Nodewebba provides six nodes, each with five parameters that can optionally be controlled by other nodes: *a* variable, *b* variable, rhythm, duration, and velocity. Each node has a GUI interface such as shown in Figure 1, supporting configuration of the LLCF, the mapping of incoming data to node parameters, and routing of MIDI output.

Figure 1. *Nodewebba* interface for a single node.

The node design supports a number of functions not addressed in the original VMCN article. For example, a "reseed" option can be toggled, so that when a node is stopped and restarted, it will re-initialize the node with the given seed value, rather than using the last state of the node. This can enable more repetitive behavior.

While LLCF is normally discussed and analyzed on the basis of its state *x* being an integer, my implementations have set *m* to 1.0, with *x* being float-point. Knowing that all state values are in the $0 \leq x < 1$ range facilitates mapping of values in the system. New with *Nodewebba* is the ability to use negative values for *a* and *b*, which can provide useful variation in available pattern types — such as an inversion of the characteristic upward curve. Also new in a node is the ability to designate a minimum and maximum value for the *a* and *b* variables. If another node's output is routed to one of these inputs, in the incoming data is mapped to the given range. It is in allowing $|a|$ in particular to be greater than 1 that more complex patterning from a node, particularly unpredictable self-similarity, is enabled.

For MIDI note output, a node can designate a musical mode, the tonic for that mode, and then a minimum and maximum index into the mode. The LLCF state-variable is then mapped to the index range. For example, in Figure 1, the state-variable is 0, which mapped to the index range of -7 to 7 will yield a -7. For a major scale with a tonic of C4, the -7 will yield a C3.

For MIDI-note output, a node can be assigned a musical mode, the tonic for that mode, and then a minimum and maximum index into the mode. The LLCF state is then mapped to the index range. For example, in Figure 1, the state is 0, which mapped to the index range of -7 to 7 will yield a -7. For a major scale with a tonic of C4, the -7 will yield a C3. While conceptually convenient for many purposes, one shortcoming of this approach is that it takes more effort to figure out the correct configuration needed to keep note output between certain values — such as range limits of target instruments. If a user wants to map to tunings, modes, or musical parameters other than those provided — or other media targets altogether — the floating-point state-value output of a node can accessed directly and mapped via an API (see 3.3 below).

Rhythm and duration are determined by scalars that are mapped to minimum and maximum integer beat values. Though "beat" was chosen as a user-friendly term, this really refers to 16th-note ticks of a master clock controlled by the global tempo setting. The scalars can either be hand-specified through the interface or driven by another node. The duration scalar can exceed 1.0, creating notes longer than the maximum rhythm value.

So quantized rhythm is foundational to *Nodewebba*. This unfortunately does preclude using *Nodewebba* to create some of the interesting non-quantized rhythmic effects that VCMNs can produce. Tuplet relationships can be established between nodes, particularly if one configures each node to have only a single rhythmic value. On the other hand, variable-length tuplet notes that form neatly aligned groupings cannot always be ensured.

Indicating a randomness range value can humanize the start time and velocity of the MIDI data. The floating-point output is not humanized, since this would make it impossible to ensure such consistent, repeatable behavior in most network configurations where rhythm is determined by node outputs.

Finally, the user can indicate a target MIDI device, channel, and patch number via menus. A "Mute" toggle allows one to let the node continue its activity while muting its MIDI output. In this case, the state-variable output continues, so the node can continue to operate as part of the network even while silent.

2.2 Matrix Interface

A matrix interface (Figure 2) allows easy, realtime configuration of the network, with source node-outputs on the top and target node-inputs on the left. Push buttons allow the matrix to be cleared or randomly populated. Presets can be stored, and sets of presets can be saved to or retrieved from disk.

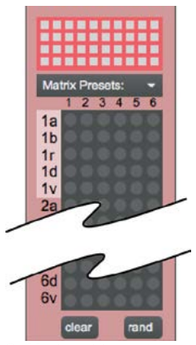


Figure 2. *Nodewebba* matrix interface.

2.3 Transport and MIDI Control

A main transport interface provides a main on/off function, external sync on/off, global tempo, and preset defining, storage and retrieval for the transport and node configuration. A MIDI Control window allows the user to assign MIDI controller inputs to the transport functions as well as to key parameters for each node: on/off; mute; reseed; a min/max; b min/max; velocity min/max.

3. IMPLEMENTATION ISSUES

3.1 Initialization and State Logic

System design solutions that support intuitive and consistent results for composers are less obvious than it might seem at first consideration. This can be demonstrated with a few example scenarios, describing the solutions established with *Nodewebba* version 0.05.

Consider a single node, seed 0, *a*=1, *b*=0.1. When starting this node, a logical expectation is that the first emitted state will be 0, not 0.1 Further, if the user later changes the seed, the next state output should be this new seed. Thus, the core iterated map enters a 'seed changed' state when the user provides a new seed. When fired, it emits this seed and then enters the 'iterating' state, where further firing results in iteration of the map.

Consider two nodes, N1 and N2. N1 controls the *a* and *b* variables of N2, and visa versa. For this to work in a consistent fashion, the firing of all nodes must occur prior to updating the nodes with the new emitted control values. Otherwise, for example, the N1 might iterate, emit its new state, and change N2's variables before N2 iterates.

To ensure repeatable results, rhythm is implemented through a clock pulse at a 16th-note rate in the given tempo. At each clock pulse, all nodes first push the last received control data into an active state. Then all nodes increment a counter. Once the counter receives enough triggers to reach the rhythm beat-count, the node fires: the iterated map is activated and its resulting state-value is emitted and sent to any targets node inputs in the network as potential future control data.

In the case of a node controlling its own rhythm and duration, the user would likely expect rhythm and duration to directly reflect the emitted state of the node. That is, if the node emits a 0, the shortest note value should ensue at that time, rather than on the next firing. Likewise, if the node emits a 1, the longer possible note value should ensure. To support this behavior, the firing step places the node in the 'ready to emit MIDI' state. After all nodes have been invited to fire, they are all then invited to emit MIDI data. If a node it ready to emit, it will calculate rhythm and duration based on most recently received control data (which might come from nodes that just fired 'now'), generate the MIDI output and set the beat counter target for next firing of the node.

In summary, then, at each metro clock *Nodewebba* executes 'update inputs' for every node to gather the latest variable-control inputs, then calls each node to update its counter and fire if ready, and then has each fired node update its rhythm and duration based on any just received control data and then emit the MIDI data.

3.2 Other Issues

Allowing negative values for *a* and *b* means that a standard modulo function no longer suffices. Instead, the function must wrap when it receives a negative number. This is implemented as follows:

$$\text{for } x \geq 0,\ f(x) = x \bmod 1$$
$$\text{for } x < 0,\ f(x) = \big(1 - (x \bmod 1)\big) \bmod 1 \quad (2)$$

The outer mod 1 in the latter formula ensures that a 1 returned by the inner mod becomes 0, thereby maintaining the expected $0 \leq x < 1$ output range. In Max, implementation is complicated by the fact that Max can return a negative zero in floating-point calculations, which will return *true* if tested to see if it is less than zero. The current solution tests instead to see if *x* is less than 0.000001.

3.3 Postprocessing and API

Though *Nodewebba* can be used as-is, many composers might wish to provide additional post processing, automation or links to other systems. This is particularly true given that no musical knowledge is embedded in the network itself, and only minimal musical knowledge (in the form of modes) is supported in the mapping. Therefore, additional logic based on specific musical intents may be required.

Post processing is implemented in a subpatcher that receives the output messages from the nodes. The MIDI notes generation and humanization already occur here. Working with the source version of Nodwebba (versus the standalone), a Max coder can easily add additional post processing functionality here without having to intercede with the lower-level node code.

Further, API hooks are provided in the form of accessible variables (Max sends and receives) and a standardized naming scheme that includes node numbers in the variable names. For example, one can control the on/off state of Node 5 by addressing a [send 5o] object, or receive the state-variable output of Node 3 with [receive 3val]. Thus a coder can interface with *Nodewebba* without needing to touch any *Nodewebba* code.

3.4 Synchronization

A Max *hostsync~* object allows a ReWire-enabled external sequencer to control the *Nodewebba* transport and clock via the ReWire protocol, preventing timing from slipping between the two programs when recording *Nodewebba* output to the sequencer.

4. APPLICATION

Extensive details regarding the use of VCMN in the composing of *Clonal Colonies* can be found elsewhere [8]. Here we will just look at a few examples of external control and post processing in that piece. In short, each instrument (flute, bass clarinet, violin, cello, piano, and gongs) was assigned to one node, since this was conceptually the simplest way to approach the work. Nonetheless, as noted in the original article, the systems still readily become too complex to treat analytically: one must use exploration and heuristics to develop compositional possibilities.

One original impulse behind the VCMN technique was the idea that some network configurations might exhibit higher-order emergent "change in the nature of change", leading to a greater sense of dramatic plateaus or even trajectories. Instead, complex networks with many feedback links tend to create "too much variety", where the output risks being perceived as exhibiting continuous change without significant-seeming patterning. In these cases, too, it can be very hard to find meaningful ways to intercede with the settings to provide a sense of musically coherent transformation of behavior. In practice, overcoming this requires explicit design or external control mechanisms, or by placing some nodes in a position of clear hierarchical control over the system, typically operating at slow rates of change.

In *Clonal Colonies*, to address the wide range of potential network configurations and the lack of predictability in results, I connected an external MIDI slider-controller box to numerous parameters of the system, allowing relatively fast exploration of behaviors and gradual development of a structured improvisation.

Post processing routines were also central to addressing aesthetic and practical technical issues. One issue with VCMN is lack of a natural phrasing mechanism. Besides the resulting risk of monotony, the lack of phrasing can be particularly problematic when writing for wind instruments, since the performer needs opportunities to breathe. Inspired by practices common in Hindustani classical music, I decided that certain notes in the mode would receive extra emphasis — in this case by trilling them and using them to end phrases. Post-processing code detected when a node generated these pitches. The code then sent the instructions to generate a trill and turned off the node. The node would then wait for a given duration — set via the external controller — before turning on again. Thus this pause-duration control essentially served as a core density control for the whole ensemble, providing an important tool for high-level shaping of dramatic form, not provided in the VCMN itself.

Several takes were then captured to a sequencer prior to editing. This served as a sketch for the work. I gave myself the creative constraint of making the captured output of the system more convincing, not through editing, but by changing its context through the addition of computer rendered sound. In this sense, I believe VCMN and *Nodewebba* may often serve well as generators of ideas and rough structures on which a composer can then elaborate, gaining creatively from the surprises generated from the system.

5. CONCLUSIONS

Nodewebba makes it considerably easier for composers to experiment with and apply both simple LLC approaches and more advanced VCMN techniques. It also facilitates live performance with VCMN. It does not significantly add to the originally proposed VCMN concept, but its development did lead to clarification of some aspects of system design for consistent and intuitive usage.

It would be ideal for *Nodewebba* to allow non-quantized rhythmic output, but this would require creation of mechanism that would allow this while also providing strict repeatability. Clearly, too, there is room to explore iterative functions other than LLCF. An ideal implementation of *Nodewebba* would allow selection of different functions. Functions that required different variable counts or don't have a fixed output range, however, would offer a significant design challenge.

Nodewebba binaries and video demonstrations are available for download at http://BatHatMedia.com/Software/Nodewebba, and the source is available at https://github.com/bbattey/NodeWebba.

6. REFERENCES

[1] C. Ames, "A Catalog of Sequence Generators: Accounting for Proximity, Pattern, Exclusion, Balance and/or Randomness," *Leonardo Music J.*, vol. 2, no. 1, pp. 55–72, 1992.

[2] P. L'Ecuyer and F. Blouin, "Linear congruential generators of order K>1," *Proc. 1988 Winter Simul. Conf.*, pp. 432–439, 1988.

[3] B. Battey, "Musical pattern generation with variable-coupled iterated map networks," *Organised Sound*, vol. 9, no. 2, pp. 137–150, 2004.

[4] W. Burt, "Algorithms, microtonality, performance: eleven musical compositions," PhD Thesis, University of Wollongong, 2007.

[5] F. M. Pérez, "Tècniques de microsampling amb generadors de congruències lineals," 2013. [Online]. Available: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/19073/6/martifrancescTFM0113memoria.pdf [Accessed: 25-Feb-2016]

[6] F. Peréz, "Speech 2 [video]," 2015. [Online]. Available: https://vimeo.com/119713106. [Accessed: 12-Feb-2016].

[7] C. Emmeche, S. Køppe, and F. Stjernfelt, "Explaining Emergence: Towards an Ontology of Levels," in *Systems Thinking, Volume 1.*, G. Midgley, Ed. London: Sage, 2003.

[8] B. Battey, "Creative Computing and the Generative Artist," *Int. J. Creat. Comput.*, vol. 2, no. 1, 2016.

Relative Sound Localization for Sources in a Haphazard Speaker Array

Neal Andersen
Department of Music and Arts Technology,
Indiana University-Purdue University Indianapolis
andersne@iupui.edu

Benjamin D. Smith
Department of Music and Arts Technology,
Indiana University-Purdue University Indianapolis
bds6@iupui.edu

ABSTRACT

A rapidly deployable, easy to use method of automatically configuring multi-channel audio systems is described. Compensating for non-ideal speaker positioning is a problem seen in immersive audio-visual art installations, home theater surround sound setups, and live concerts. Manual configuration requires expertise and time, while automatic methods promise to reduce these costs, enabling quick and easy setup and operation. Ideally the system should outperform a human in aural sound source localization. A naïve method is proposed and paired software is evaluated aiming to cut down on setup time, use readily available hardware, and enable satisfactory multi-channel spatialization and sound-source localization.

1. HAPHAZARD ARRAYS

A haphazard speaker array involves any number of speakers (more than 2), placed in a space with little regard to precise alignment, orientation, or positioning. Unlike speaker grids or uniform array setups, the haphazard array is created at the whims of the user, potentially responding to constraints of the environment to guide placement (such as limitations in mounting, positioning, and cable lengths), or to take advantage of unique acoustics of a given installation space. Further, the haphazard array may use any mix of speakers with significantly different acoustic characteristics. While a conventional, uniform array focuses on pristine, reproducible audio, the haphazard model seeks to exploit unique elements of a given installation, equipment, and space.

The haphazard array presents a complex system with potential acoustic richness unique to each setup. The array also works within each environment it is setup in, providing a further layer of acoustic interaction that makes each configuration unique. A primary goal of haphazard arrays is a quick and inexpensive setup, using equipment that is on hand and spending a minimum of time calibrating the system.

The goal of this project is to research and define methods of working with haphazard arrays that make their complex nature transparent to the user. Ideally the capabilities of the system should be easy to use, leveraging current live mixing practices. The user should not be burdened with learning the particulars of the array's configuration, rather they should be able to use a uniform panning interface (sec. 3) which hides the complexities of the array. Similarly the setup and configuration of the system should support rapid deployment and minimal time from connection to use.

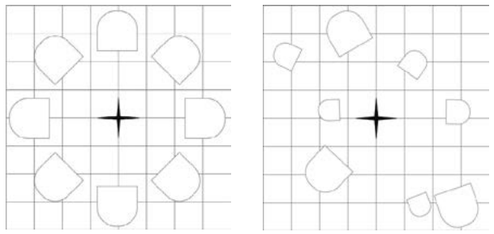


Figure 1. Fixed-Speaker Array (Left); Haphazard-Speaker Array (Right)

2. BACKGROUND/CONCEPT

Researchers in both acoustics and robotics address automatic identification of speaker array characteristics, such as sound-source/speaker location and frequency responses. The goal of this project is to provide a single point of interaction for a user to mix one or more tracks of audio within the array's acoustic space. Given a fixed uniform array (Fig. 1, left), the controls typically take the form of a panning potentiometer or digital dial to mix the source audio between output channels. This same model can be extended to work across non-uniform arrays (Fig. 1, right) if the characteristics of the setup can be accurately mapped.

2.1 Auditory Localization Issues

Describing the speaker locations and characteristics is closely related to research in robotic audition which looks at building systems to isolate and locate sound sources to inform robot functionality. Popular robotic approaches are based on models of human hearing, and typically start with two or more microphones mounted in opposed directions, performing calculations based on inter-aural intensity difference [6] and time difference of arrival [3] (i.e. the difference in time between a sound's arrival at each 'ear'). The accuracy of these systems (typically within centimeters for nearby sounds) greatly improves with the employ of more than two microphones, allowing the robot to assess sounds in a 3-dimensional field [8].

Tests conducted with human subjects show a wide range of error in localizing depending on the frequency and angle in which the sound source is played. In one study [1], test subjects displayed horizontal angle accuracy between 8.5–13 degrees in testing audio along the horizontal plane without visual cues.

The minimum audible angle of humans for the horizontal plane has improved accuracy if the sound is in front of the listener and the test tone is brief [5]. This optimized scenario displayed accuracy between 2–3.5 degrees. However as the sounds moved to the side and behind the head, the error reached up to 20 degrees.

Measuring the perceived distance from human listeners is almost incalculable, as distance is considered to be lateralized, or processed internally as opposed to localized from an external cue. [8] To accurately localize distance from the arrival time of a sound source in a human, there needs to be some kind of non-auditory sensory feedback. [9]

2.2 Speaker Systems

Another similar problem is the automatic calibration of home surround sound systems, which are commonly setup in a less than ideal fashion. Using a microphone array these approaches play test tones through all the speakers in the setup in order to identify the particulars of the setup, acoustic characteristics of the room, and listener's sitting location [2, 7]. Time difference of arrival is the primary approach taken for speaker identification. They anecdotally report speaker location accuracy to several centimeters, in spaces no larger than 9 feet square.

The performance requirements of a haphazard array are based on the discriminatory ability of the people who will be experiencing it. Thus human auditory accuracy defines operating success of a calibration system.

Evaluation of panning algorithms with human participants showed a consistent average accuracy across all models of 10 degrees [4]. However every test showed many individual errors of up to 45 degrees, regardless of panning algorithm employed.

3. LOCALIZATION FACTORS

In order to create a panning interface the characteristics of the array have to be measured and analyzed to build a virtual map of the array. This can be accomplished manually, with a user entering data for each speaker into the system, but this is cumbersome and expensive (in terms of time), requires expertise, and works against the goals of having a quickly usable system. Automating the configuration of the system is the preferred solution and involves analyzing the acoustic space for the following information:

- The position of each speaker,
- The relative loudness of each speaker,
- The relative frequency response of each speaker.

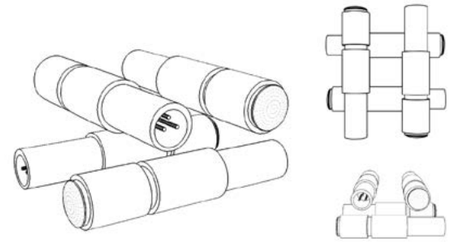


Figure 2. Microphone X-Y grid

The accuracy of this system needs to be more accurate than human listeners in order to convincingly spatialize sounds. With the final aim of informing a real-time panning system, such as a 360° dial, for live use we prefer a simpler, naïve approach.

The proposed system analyzes the speaker array using a 4-way X grid of microphones (see Fig. 2) setup in the nominal center of the space (Fig. 3). A frequency rich[1] test tone is played through each speaker in turn and recorded through the four microphones. These recordings are then analyzed and the source location estimation is performed. This information is then used to inform a panning interface.

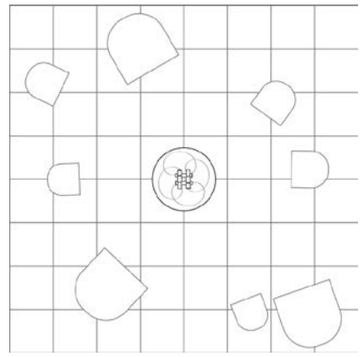


Figure 3. Diagram of microphone within array

4. MODEL AND MAXFORLIVE OBJECT

Before other aspects can be analyzed the overall latency of the audio system must be measured (i.e. the time from sound output to the return of that sound through a microphone). We accomplish this by holding a microphone on the grill of a speaker, playing a test tone through it, and calculating the interval between onsets by looking at signal threshold crossings. This latency time is used as a

[1] Tests with a straight sine tone and tests with various colors of noise resulted in widely anomalous estimations across different speaker positions. Tones with many frequencies (such as those of an acoustic instrument or voice) were found to be more consistent.

baseline to estimate speaker distances based on tone times of arrival.

Estimation of speaker position is performed using brute force loudness estimations rather than inter-aural timing differences. Given the priorities of speed and robustness this method is able to take advantage of the pickup patterns of commonly available unidirectional microphones (such as the Shure SM57, see fig. 4).

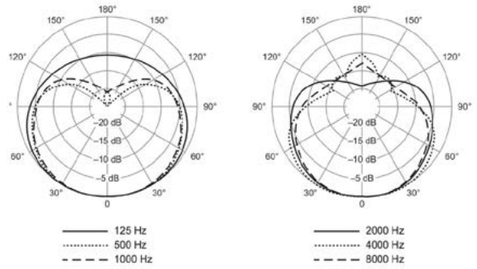


Figure 4. Shure SM57 Polar Pattern

Theoretically simple triangulation of the speaker positions (from the decibel level captured over the four mic grid) would be possible with ideally isolated microphones with precise pickup patterns. Commonly available microphones pickup much more than 90° and have non-linear input responses (i.e. discontinuous around the polar pattern of the microphone). However, given a set of four identical microphones (within the specifications of the manufacturer) it is possible to deduce position through cancellation. That is, as a sound source moves along the axis of two opposed microphones the change in measured intensity will vary in a consistent fashion. The decibel level (D) is calculated as the root mean square of one-second of audio samples ($x_{1:N}$) from one microphone.

$$D = \sqrt{\frac{(x_1^2 + x_2^2 + \ldots + x_n^2)}{n}} \quad (1)$$

With two matched uni-polar microphones facing in opposite directions the location of a sound source along the axis correlates with the signed difference between the measured input levels. This is repeated for the same sound source along the laterally perpendicular axis giving a Cartesian estimation of the source (speaker) location (axis x, y with input levels 1, 2). This allows an estimation of the angle to the sound source from the center microphone grid:

$$\theta = \tan^{-1}\left(\frac{y_1 - y_2}{x_1 - x_2}\right) \quad (2)$$

The distance that can be calculated from these measurements will be highly influenced by the characteristics of the microphones (for example, hyper-cardioid microphones pickup effectively at 90° off-axis and this would make sound sources seem closer than they are). Using the amplitude (and the theoretical reduction in decibels over distance) recorded by the microphones as an estimation of distance is similarly influenced by reflections and resonances of the environment. We found a simple time of arrival measurement performs consistently across speakers and with minimal environmental sensitivity.

The distance to each speaker is estimated based on the latency between the initiation of the test tone and the time of arrival (τ) of the same tone at the microphone grid. Removing the known system latency (z) gives a time indicating distance (d) to the speaker, using the known speed of sound at sea level (C) of 1,126 ft./second.

$$d = C(\tau - z) \quad (3)$$

With an estimation of all of the speaker locations, panning between speakers is accomplished with a software interface, implemented for Ableton Live as a MaxforLive device (see Fig. 5).

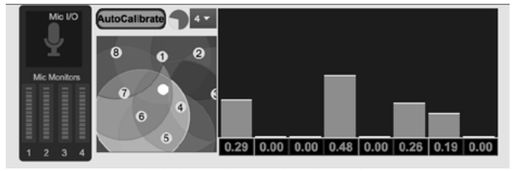


Figure 5. Multichannel panning interface prototype.

There are special configuration requirements for the Ableton Live session file to work properly with the device. The latest version of the device is implemented to Live by corresponding the estimated distance and angle data for each speaker to set the level of each Return Track, which is determined by how many external outputs (speakers) you are using.

After calibration, the method of control takes place on each individual track or group. The current control interface utilizes a node object as the main point of control. Upon dragging the unnumbered node closer or further from the numbered nodes (speakers) the levels will rise and fall accordingly in real time (Fig. 5).

5. TESTING

To evaluate the proposed system 800+ data points were recorded at around 240 different speaker positions. Four Shure SM57 microphones were used for the test grid and the same hardware was used for all data measurements. The tests were performed in a large (20x30 ft.), acoustically treated room with a minimum of sound-reflective surfaces and background noise. At each speaker position the location in the room was measured relative to the center of the microphone grid, and a 4-channel recording was captured of the test tone playback. This recording was processed as described above and the angle and distance to the speaker was estimated. The performance is characterized in Table 1 by error in estimated distance, error in estimated angle, and the magnitude of the distance error (i.e. error divided by measured distance to show scale of error). Figure 6 shows the error in angle in degrees across all data points.

The error in angle measurement appears to be independent of actual distance to the speaker (within the tested 2-30 foot range), and does not correlate with the distance to the speaker, as shown in figure 7 (error in angle, in degrees, graphed over distance to speaker).

Error:	Mean	Standard Dev.	Max.
Angle to speaker	4.95°	4.45°	23.09°
Distance to speaker	1.15 ft.	1.60 ft.	9.25 ft.
Magnitude of distance error	10.75%	13.07%	135.25%

Table 1. Estimation error.

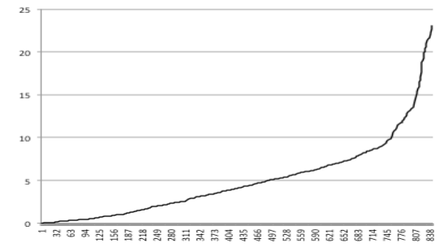


Figure 6. Error in angle (in degrees) across all test data.

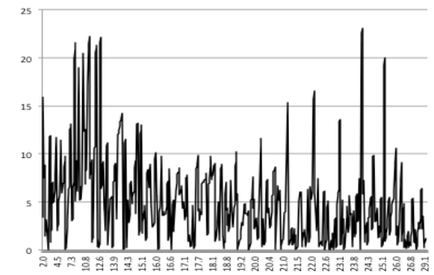


Figure 7. Error in angle over distance (in feet).

This data shows that the system can estimate the angle to a speaker within 4.45 degrees, and the distance to the speaker within 1.6 feet. The error in distance does not strongly correlate with actual distance, (i.e. the error does not increase with actual distance). Likewise the error in angle does not significantly correlate with distance (i.e. the system performs independent of actual distance).

While solutions such as [2, 3, 6, 7, 8] are able to locate sound sources with an accuracy of centimeters given smaller spaces, our system works with reasonable accuracy at a larger scale.

6. CONCLUSIONS

Considering the goal of supporting rapid deployment of speakers and minimal setup time, the software achieves a 2 second calculation time for each speaker could theoretically detect the speaker location of an 8-source array within 16 – 32 seconds depending on the level of desired accuracy. The accuracy of the estimation performs roughly twice as well as human audition, suggesting that the resulting panning system is accurate enough to satisfy a listener's discriminatory ability. Future studies with human participants will determine if practical application is satisfactory for real world use.

Frequency response of each speaker is an important characteristic in building an accurate system. However the current model does not address this aspect. Performing spectral analyses of the test tone playing through each speaker, de-convolved with the tone, should identify the frequency response of each individual speaker. This can then be used to inform an EQ calibration to ensure a uniform audio image across the entire array.

Future goals include putting this software into practice in a full 8-speaker setup. In this environment the accuracy of human listeners standing in the same position as our microphone array can be tested. Further, use tests can be conducted to compare panning algorithms with different practical sound material.

Extending the current model to enable speaker elevation detection could be accomplished through reconfiguration to a tetrahedral microphone grid (i.e. 4 microphones facing out in a pyramid formation).

7. REFERENCES

[1] Carlile, Simon. "Auditory space." *Virtual auditory space: Generation and applications*. Springer Berlin Heidelberg, 1996. 1-25.

[2] Fejzo, Zoran, and James D Johnston. 2011. "DTS Multichannel Audio Playback System: Characterization and Correction." In Audio Engineering Society.

[3] Hu, Jwu-Sheng, Chen-Yu Chan, Cheng-Kang Wang, Ming-Tang Lee, and Ching-Yi Kuo. "Simultaneous localization of a mobile robot and multiple sound sources using a microphone array." *Advanced Robotics* 25, no. 1-2 (2011): 135-152.

[4] Kostadinov, Dimitar, Joshua D Reiss, and Valeri Mladenov. 2010. "Evaluation of Distance Based Amplitude Panning for Spatial Audio." In *ICASSP*, pp. 285-288..

[5] Makous, James C., and John C. Middlebrooks. "Two‐dimensional sound localization by human listeners." *The journal of the Acoustical Society of America* 87.5 (1990): 2188-2200.

[6] Nakadai, Kazuhiro, Hiroshi G. Okuno, and Hiroaki Kitano. "Real-time sound source localization and separation for robot audition." In *INTERSPEECH*. 2002.

[7] Shi, Guangji, Martin Walsh, and Edward Stein. 2014. "Spatial Calibration of Surround Sound Systems Including Listener Position Estimation." In Audio Engineering Society.

[8] Valin, Jean-Marc, François Michaud, Jean Rouat, and Dominic Létourneau. "Robust sound source localization using a microphone array on a mobile robot." In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, pp. 1228-1233. IEEE, 2003.

[9] Zwiers, M., Al Van Opstal, and J. Cruysberg. "Two-dimensional sound-localization behavior of early-blind humans." *Experimental Brain Research*140.2 (2001): 206-222.

AIIS: An Intelligent Improvisational System

Steven Leffue
University of California, San Diego
stevenleffue@gmail.com

Grady Kestler
University of California, San Diego
gradykestler@gmail.com

ABSTRACT

The modern use of electronic sound in live performance (whether by instrument or composed processing) has continued to give rise to new explorations in its implementation. With the construction of Aiis, we sought to build an interactive performance system which allows for musical improvisation between a live performer and a computer generated sound world based on feedback between these components. Implemented in Pure Data, the system's micro and macro decisions generate a programmable musical "personality" derived from probabilistic measures in reaction to audio input. The system's flexibility allows factors to be modified in order to make wholly new and original musical personalities.

1. INTRODUCTION

1.1 Musical Influences

The genre of electroacoustic music allows for the exploration and augmentation of sonic worlds which transcend the limitations of acoustic instruments. A key component of the genre is the ability to create musical structures which draw fundamental features from digital processes such as algorithmic composition, mathematical modeling, and interactive systems [1, 2].

Contemporary performers are also making use of electronic sounds in genres across all genres. From playing through effects to using electronic instruments in improvisation, this marriage of instrumentalist with electronic music source offers countless approaches. Contemporary of these collaborations can be seen in the music of Peter Evans Quintet, Evan Parker's work with the Electroacoustic Ensemble, or the recent pairing of Merzbow, Mats Gustafsson, Thurston Moore and Balzs Pandi.

1.2 Intelligent Systems

Against this backdrop, our initial research centered around a few guiding questions. First, can the ultimate hallmarks of humanity, those being creativity and decision making, be digitally replicated convincingly enough to power collaborations with a live performer? Second, through analysis, can a specific performer's musical personality, otherwise envisioned as their reaction to stimuli, be conceived as a collection of probabilities and thereby replicated?

In the 1980's, Rodney Brooks put forth a new foundation for artificial intelligence known as subsumption architecture. This foundation was a representation of the reactive paradigm. Before Brooks' model, necessary and sufficient properties of machine intelligence were thought to consist of three crucial components: sensing, planning, and acting. It was during the planning stage that a machine would analyze sensor data to generate symbolic representations of objects in the world in order to react accordingly [3]. The reactive paradigm, however, eliminates this intermediate phase and simplifies the acting stage. In lieu of the planning stage, the reactive paradigm implements substantially more sensing and reacting behaviors which work together in a non-hierarchical fashion to generate intelligent behavior.

Aiis is the first instantiation of an interactive system which attempts to address our questions by implementing Brooks' reactive paradigm architecture. It seeks to provide a continuously creative system which is able to interact with a performer by responding to stimuli and creating its own content. Included in its programming are features which react to stimuli in an improvisatory and musical fashion.

2. BIG PICTURE

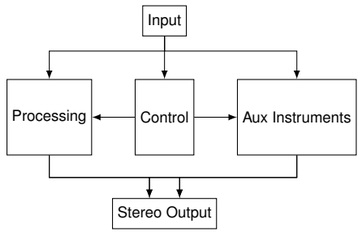


Figure 1. Over arching flow of the machine. Input from a performer is sent to the *processing*, *control*, and *auxiliary instrument* modules to be manipulated. The stereo output is a sum of the outputs from processing module and the auxiliary instruments

2.1 Processing

In order to create the sound world, live audio is routed to four separate channels within the *Processing* module. Each of these channels pitch shifts the performer's input signal by a specified number of semitones to extend the range of a two and half octave instrument to almost eight octaves.

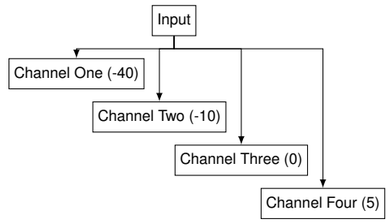


Figure 2. Abstracted view of the processing module. Input from the performer is sent to four distinct channels to be processed for the sonic environment. Each channel pitch shifts the audio input by the number of semitones specified in parenthesis.

Though independent of each other, the channels all contain an identical effect chain for processing the live audio as illustrated in Figure 3. Each of these effects also contains anywhere from one to four parameters which can be set independently by *Control* thus further expanding the channel's versatility.

Figure 3. Chain of effects through which the performer's audio is processed.

2.2 Aux Instruments

After testing and listening to early versions of this system, it seemed the overall texture still needed a boost of saturation. The purpose of the *Auxiliary Instruments* module is to create this soft bedding on top of which the rest of the sonic environment of the piece develops. This is achieved with the addition of two instruments: *Sub-sound* and *Glisser*. The former is overdriven white noise modulated through a combination of delay line, low-pass filter, and hi-pass filter. The latter, embedded within each sub-channel of the *Processing* module, is tapped after channel pitch shifting and performs granular synthesis on a section of captured signal. It will then glissando to a subsequent capture via waveform and pitch to create a soft, flowing, tonal contrast to the saturated colors of its partner.

2.3 Control

The *Control* module establishes the mechanisms for controlling a number of parameters within the system. It accomplishes this goal on three distinct levels. In its most basic function, referred to below as Non Reactive controls (NR), it creates a perpetually original audio output by randomly altering effect parameters in the channels of the *Processing* module. On the Small Scale Reactive level (SSR), analyses of the performer's audio overtly influence a smaller more select group of factors within the *Processing* module's effect parameters. Finally, based on analysis of audio, Control effects another set of parameters we will refer to as Large Scale Reactive (LSR) which are intended to aid in the creation of macro structures, musical form, and time.

2.3.1 Non Reactive Controls

The lowest level of controls operates on the smallest objects: the effect parameters. At this level, there is no reaction from the system to the performer, rather, attributes of the non-repetitive nature of the output are constantly generated. This is where the sound world gains its richness and depth.

We quickly found that while random processes can easily accomplish this goal, implementing a simple random operator would only succeed in altering the parameters at a periodic rate. The solution, named "double random", is a simple tool which generates random numbers at random intervals of time less than a given parameter. This tool is used exclusively in the control of every effect parameter found in the *Processing* module. For this instantiation of *Aiis*, the operation was set to occur at random time intervals between zero and eight seconds.

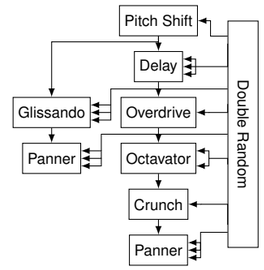


Figure 4. Non reactive control flow on a micro level. The effect parameters, indicated by the arrows on the right side of each, are controlled randomly through the "double random" object in Figure **??**

2.3.2 Small Scale Reactive Controls

The next level up from NR controls are the SSR controls. SSR analyzes amplitude and note rate from audio input in order to effect parameters in the *Processing* module. For example, when a performer increases in volume the system senses this and may instigate a "freeze" on the channel's delay effect (essentially turning off the NR controls managing parameters). SSR controls output parameters as indicated in Figure 5

Perhaps due to their reactive nature, but certainly as departures from the normative functioning of the piece, these controls are more pronounced within the generated sonic environment of the machine. As a result performers are more aware of these developments in what can be imagined as the middle ground of musical thought and will react accordingly. In this way we can think of the SSR controls as generating musical phrases. Though important in construction of the

musical world moment to moment, these controls are not as crucial to the macro level musical structures managed by the LSR controls.

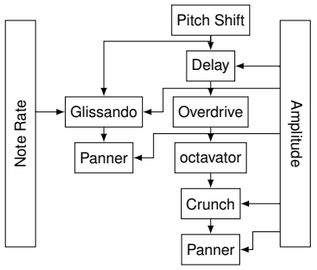


Figure 5. Small scale reactive controls. Certain effect parameters are controlled by the amplitude of the performer, while some are controlled by the note rate.

2.3.3 Large Scale Reactive Controls

There are a few behaviors that an improvisor or performer exhibit which we felt were necessary to replicate on the macro level. LSR controls were implemented to enable the system to generate musical structures in collaboration with a performer. First, we enable the system to ”freeze” in a given state for an elongated period of time. Second, since a live performer does not always react instantaneously to their partner, we implemented a variable reaction time to input. Last, we created a sensitivity to musical structure vis a vis saturation and complexity over time. These controls offer a variety of long term, clearly audible choices concerning the structure of a performance.

The system's analysis of the rate of notes controls the turning on and off of the NR control of all channel delays and pitch shifts essentially ”freezing” the output in a given state. Unlike the SSR controls, the large scale note rate reaction will freeze or re-instantiate the delay for every channel simultaneously. The same process is incorporated in regards to pitch shifting; every channel yields to this control. Note

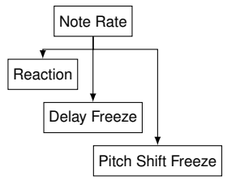


Figure 6. Large scale reactive control. The performers note rate will affect the reaction time of the computer as well as whether or not to freeze the pitch shift and delay for all channels.

rate also controls the ”reaction” parameter. This parameter was created to introduce a human-like reaction time to input from the performer. This parameter uses a bell shaped, gaussian distribution (with a variable mean and pre-programmed variance) to represent an expected reaction time and the intuitive experimentation of an individual which may deviate from this norm. The variable mean is controlled by the note rate of the performer which allows the computer to react, in general, more rapidly if the performer is playing faster and more slowly if the performer is playing longer notes.

At its core, the system is intended to react to the performer and generate sounds that would contribute to the musical world, which would then be interpreted and reacted to by the performer. This generates an improvisational feedback circuit. If the circuit breaks, if one or the other entities refuses to listen and react, we may lose a sense of musical structure.

Amplitude controls the most perceptible macro events generated by the machine: entire channels of the *Processing* module turning on and off. This parameter contains four separate levels within itself, each one dependent on input amplitude. If the performer is at his or her softest level, the machine will only turn one channel on at a time. Likewise, at the second softest volume, the machine will turn either one or two channels on at a time. The same idea applies to the third level, but at the fourth level (the performers absolute loudest level), all channels will be opened.

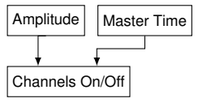


Figure 7. Large scale reactive control from performer's amplitude and *Master Time*. The amplitude will decide how many channels to turn on at a given time, while the *Master Time* decides when to do this.

The final addition to this piece was made after months of trials. In performing with the system, the lack of musical pulse became apparent. This inhibited the ability to develop structure within the piece and provided an uncanny valley for musical action. In almost every musical piece, except those which feature an abstraction of time, a pulse can be distinguished and followed by both the performers and the audience. Whereas this pulse is commonly represented by bars and measures, in free improvisation it often takes the form of long term macro beats or ”breaths”.

”Master Time” was implemented to portray these large scale musical pulses within the overarching structure of the piece. While the amplitude of the performer influences *how many Processing* channels are present, ”Master Time” controls *when* they are. The result of this is heard in the system's ability to maintain large scale musical sections punctuated by macro periodicity despite the random nature of the aural world developed by the NR and SSR controls. After experimenting with various time intervals, the current version sets ”Master Time” was set to five seconds.

3. PERSONALITY

Up to this point, we have discussed each level of control and how they react to the performer, but another step was implemented to make controls contribute to the ”personality” of the

Effect	Off	50% Mix	100% Mix
Delay	10	45	45
Overdrive	10	45	45
Octavator	72	2	16
Panner	50	0	50

Table 1. The percent chance for the effect to turn on at 50% mix or 100% mix, or for the effect to turn off. These probability measures account for the machine's creativity. Every effect parameter is decided on by probabilities.

system. The term personality (or predilection), in this sense, means the ability to make decisions with respect to the sonic environment or large scale musical structures and the probability with which those decisions are made. If this personality were absent, each of the effect parameters and controls described above would mirror a one-to-one mapping (e.g., if the performer is loud, the delays will freeze). When considering replication of a given personality, if we consider a performer's predilection to be the probability of their reaction to a certain stimulus, then we must also implement the opposite reaction as a type of musical ”creativity”.

```
if (amplitude > baseline_amplitdue + 15)
      output 1 with probability .2;
else if (amplitude < baseline_amplitude - 5)
      output 0 with probability .03;
```

Figure 8. Pseudocode example to assign probabilities to small and large scale reactive controls.

Probability measures were therefore introduced in order to account for this notion of predilection vs. experimentation. Table 1 illustrates examples of how the percentage of an effect in the audio mix is controlled by a probability as well as the ”double random” object. Each time ”double random” outputs a bang, the machine filters that bang as to whether it will turn off the effect or turn on the effect at 50% or 100% mix with unique probabilities. From the table, it is observed that the octavator has a very low chance of turning on while the delay and overdrive have a much higher chance of doing so. This idea of implementing creativity as a combination of probabilistic measures is also applied to SSR and LSR controls.

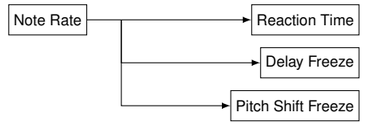


Figure 9. Reactive paradigm for the note rate sensors. The machine reacts to the note rate sensors in each of the three behaviors on the right. Each behavior is independent of the other.

The implementation of subsumption architecture and the reactive paradigm largely contributes to the system's success in generating musical form. On the left side of Figure 9, the sensing mechanism is the performer's note rate while the right side illustrates each of the possible reactions. These behaviors maintain the machine/performer circuitry that is crucial to generating music. Because each is independent of each other, there is no hierarchy of decisions, our system is made free to interpret and react to the performer. Similarly, Figure 10 illustrates the reactive paradigm as it pertains to the amplitude sensors of the system. Together, the behaviors exhibited in each of these figures underlie the musical decisions conveyed by our system.

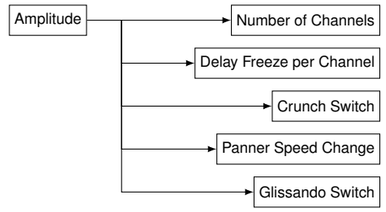


Figure 10. Reactive paradigm for the amplitude sensors. The machine reacts to the amplitude sensors in each of the five behaviors on the right. Each behavior is independent of the other.

4. CONCLUSION

The reactive and non-reactive nature of the controls, coupled with the probabilistic personality measures, creates a improvisational tool which enables musical experimentation and ever evolving structure. The different levels of control allow for the production and maintenance of the sonic environment, as well as larger structural gestures. If a performer is not constantly aware of the aural world generated by the machine, the resulting music lacks congruence in its textural qualities. If the performer refuses to acknowledge the machine's larger scale structural changes, then the musical form will become similarly unconvincing. This method of improvisation lends itself to further exploration in other creative endeavors including movement, visual arts, performance art and theater.

5. REFERENCES

[1] G. Lewis, “Too Many Notes: Computers, Complexity and Culture in Voyager,” *Leonardo Music Journal*, vol. 10, pp. 33–39, 2000.

[2] R. Rowe, *Interactive Music Systems: Machine Listening and Composing*. MIT Press, 1993.

[3] R. Murphy, *Introduction to AI Robotics*. MIT Press, 2000.

RackFX: A Cloud-Based Solution for Analog Signal Processing

Sean Peuquet
Ludic Sound
Denver, CO, United States
seanpeuquet@ludicsound.com

David Jones
RackFX
Berthoud, CO, United States
drj@rackfx.com

ABSTRACT

This paper introduces the RackFX platform as an alternative to using digital plugins that emulate analog hardware devices (plugin modeling) as a way to incorporate analog sound characteristics into computer-based music production. RackFX technology provides digital access to actual analog hardware devices. Given existing technological, social, and perceptual tensions between digital audio workstation-based effects plugins and outboard analog processing units, the RackFX platform provides a cloud-based solution. The platform is presented as a way for a community of internet users to interface directly with off-site analog hardware for the purposes of signal processing. A technical overview of the platform is provided, which outlines user experience and various server and on-site robotic processes that ultimately support the return of an analog-effected digital audio file to the user for each job processing request. Specifics regarding robotic control of analog devices and how the digital audio is handled and routed through the system on the device-side (on-site) is paid particular attention. Finally, the social implications of using the platform are discussed, regarding the cultivation of a community of users with unprecedented and affordable access to analog gear, as a new way to leverage digital technology toward the democratization of music production tools and techniques.

1. INTRODUCTION

The fetishization of analog audio recording, production, and reproduction technology shows no sign of abating. In the large and ever expanding field of music technology, analog hardware continues to be associated with musically desirable psychoacoustic descriptors, most notably 'warmth'. While the term warmth is strongly correlated to the acoustic phenomena of harmonic distortion and high frequency roll-off, the idiosyncratic production and (inter)subjective perception of analog warmth poses interesting problems for the computer musician.

Digital music technology replicates, processes, and stores audio exactly according to its software programming and the hardware limitations of the computer the code runs on. Which is to say, digital music is (barring any hardware stability issues) deterministic–from the moment directly after analog to digital conversion until the moment the signal is converted back to analog for sound reinforcement. Modeling the effect of psychoacoustic warmth using digital signal processing (DSP) techniques thus poses a hierarchical problem regarding the accurate representation of sound; no longer is the mere capture and digital representation of an analog signal at issue, but rather the problem concerns the capture and representation of how that analog signal was produced, which necessarily entails some degree of indeterminacy. As Karjalainen and Pakarinen decribe, "virtual analog modeling seems straighforward but is found demanding due to the nonlinearities and parametric variation in the analog domain." [1] The desired perceptual excess of an analog processed signal, it's warmth, is largely a direct result of the physical components of the analog system, their unpredictability and imperfection. In this respect, the modeling of analog effects (warmth correlates) using DSP is also closely related to synthesis, specifically physical modeling synthesis.

While the modeling approach has led to great successes and a burgeouning marketplace for software instruments and analog modeled plugins alike [2], there remains both a precision problem and a perception problem regarding the refinement and accuracy of our models. The question remains: what physical interactions are necessary to model and to what degree of accuracy– sufficient to overcome the just noticable difference (JND) in respect to some analog reference point? Despite Julius O. Smith's 1996 pronoucement (regarding synthesis) that, "we appear to be approaching *parity* between real and virtual acoustic instruments," [3] we are twenty years on and it appears that the *lack* of parity is increasily what structures both the popular discourse and commecial reality of music recording and production. The cello has yet to be fully replaced, in the same way that people who actually have access to a vintage Fairchild 670 would claim that all attempts to emulate the device as a digital plugin have failed. So despite the ease and accessibility of plugin emulators, actual vintage analog hardware processing units remain the gold-standard.

Counter to the prevailing trend of digitally modeling analog processes that yield the sensuous qualities of sonic warmth, the authors have sought to simply digitize access to the analog components and processing itself. The RackFX platform is essentially a "straight from the horse's mouth approach" to analog signal processing. While the idea of enabling distributed access to physical acoustic resources is not without precedent (see the Silophone [4] or the the MIT Responsive Environments Group's "Patchwerk" web interface to Joe Paradiso's massive modular synth [5]), the RackFX platform is a uniquely scaleable and flexible solution with potentially longer-term consequences and implications. The technology was conceived of by David Jones, and developed by Jones and Sean Peuquet across much of 2015. The platform will be live for beta users starting May 14, 2016. Across the rest of this paper, the RackFX platform will be presented as both a technological solution and a paradigm shift regarding issues of access, affordability, and quality that govern the viability of signal processing using analog hardware devices.

2. TECHNICAL OVERVIEW

2.1 User Web-App Experience

The RackFX platform begins with a community of internet users. By creating an account and logging into the RackFX web-app (http://rackfx.com), each user is presented with a "dashboard" listing previously completed processing requests or 'jobs' and a drop area for uploading a digital audio file in .wav or .aif format (sampling rate and bit depth of the uploaded file are entirely flexible). See Figure 1 below for a screenshot of the current dashboard interface layout for any given user.

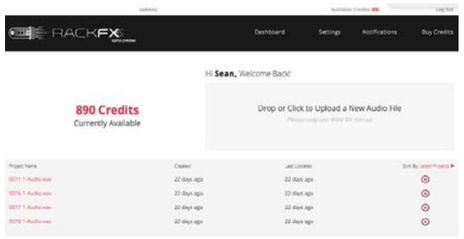


Figure 1. RackFX user dashboard

Once the user has uploaded a .wav file, she will be taken to a page displaying the waveform of the uploaded file, with an opportunity to play it back. At this point the user can decide to "add processing" (see Figure 2).

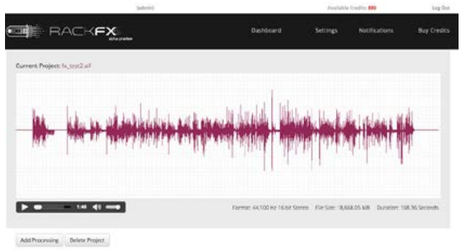


Figure 2. RackFX uploaded audio file waveform

Once the user decides to add processing to the uploaded digital audio file, she will be presented with a page detailing the devices that are currently hooked up to the RackFX system and listing which devices are currently online or active (see Figure 3).

The user selects an available analog device and is taken to a device-specific page to set parameters for how that device will process their signal. Parameters are unique to each device and so the available sliders on the web interface reflect what is available given the particular configuration of hardware knobs, sliders, buttons, etc. The user sets the desired parameters and then clicks "process audio" (see Figure 4).

Figure 3. RackFX uploaded audio file waveform

Figure 4. RackFX uploaded audio file waveform

Once the process audio button has been clicked, the user is taken back to a page showing their uploaded waveform (audio file to be processed) with a message dialog pane to the bottom right of the waveform reporting the status of the audio processing. The first step in processing is to add the processing job to a queue. (see Figure 5). The queue is a node.js application running on the RackFX server that handles the scheduling of all requests for processing received from the internet.

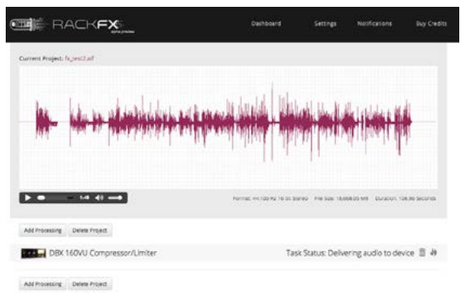


Figure 5. RackFX uploaded audio file waveform

Once the user's file has been processed and uploaded to

the server as a new file, having waited only for the device to become available (if another user is currently active) and for the audio to process in realtime, the RackFX current project page will update, show the new file's waveform, and provide a download link to the analog-effected digital file.

2.2 The RackFX Platform Behind the Scenes

On the server-side for any RackFX processing job, the web-app proxy receives all internet requests and forwards them into a web-app cluster written in node.js. The web-app cluster interfaces with Amazon S3 for storage and a MySQL database handling all site data. The queue software, a light weight file-based JavaScript Object Notation (JSON) queuing service, runs on the web-app cluster. When a job is submitted for processing, the queue schedules it, waits for the targeted analog device to become available and then passes each job, when ready, to be processed one at a time. The queue passes the job specifics to a Messaging Application Programming Interface (MAPI) cluster (also written in node.js), which messages the Machine Device Controller (MDC) software running on an OSX machine in close physical proximity to the actual analog hardware units.

At this point, the processing request has moved from the server (web-app) to the actual device-side of the processing system. The MDC (also written in node.js) orchestrates the actual processing of the (still) digital audio in the following order: (1) identify the device selected for processing; (2) switch on electrical power to the given analog hardware unit and an arduino interfacing with the device using device-specific robotic components; (3) pass device parameters to the arduino (using Johnny-Five, a JavaScript robotics module for node.js); (4) wait for the robotics to physically interact with the analog hardware device and set all parameters; and (5) spawn a Cycling74 Max application (that we call "max-io") that handles the realtime digital audio playback and capture of the analog-effected signal. Once the roundtrip signal i/o is complete, max-io tells the MDC that the new file has been written, the MDC cleans up: uploads the new file to the server, signals completion, and shuts all on-site devices down. A visual overview of the whole RackFX system, including major components and signal flow, is shown in Figure 6.

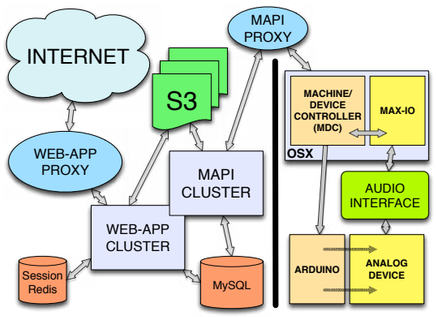


Figure 6. RackFX system overview

3. ROBOTIC DEVICE INTERFACE SPECIFICS

In order to maximize the automatization of the interaction between the user (client-side) and the analog device (server-side), RackFX aims to outfit each analog device with custom robotics that physically interact with the device's particular control panel. Various models of stepper motors, acctuators, and sensors combine to create each hardware interface between machine and device. These robotic components are controlled using a dedicated arduino board for each device (see Figure 7).

Figure 7. Robotic hardware assembly mounted on a Fender Princeton Reverb amp

Each arduino is loaded with the Advanced Firmata firmware (a protocol for communicating with microcontrollers from software on a host computer) and addressed by its host computer using the JavaScript client library Johnny-Five (J5), a robotics and Internet of Things programming framework. The node.js MDC loads the J5 module to enable communication between the MDC and each device. When the MDC goes to process a job (the next job in the queue), the program identifies the arduino associated with the specified analog device, instructs all stepper motors to reset (one at a time) by (over-)turning all knobs counter-clockwise a full rotation to ensure the analog device potentiometers are set to zero. The MDC then instructs each stepper to turn a certain number of steps commensurate to the parameter setting specified by the user through the web-app. (Maximum and minimum step values are tuned in relation to each physical parameter setting for each device, as part of the configuration process.) After a short delay to ensure all parameters are set, the MDC communicates with max-io to commence audio processing.

4. AUDIO HANDLING SPECIFICS

The Max applictation, named max-io in this project, handles all of the digital audio playback, routing in and out of the outboard analog device, and the digital re-capture of the processed audio. Max-io is designed to be as transparent as possible regarding the digital source and returned object for each processing job. Furthermore, communication between the Machine Device Controller (MDC), which orchestrates the processing of each job, and max-io follows a very specific messaging protocol using Open Sound Control (OSC) messages.

Max-io requires the MDC to provide seven parameters for each job to ensure successful completion. They are as follows: (1) number of channels for the audio signal, (2) file path to the digital input file, (3) file path for the analog-effected output file, (4) sample format (int8 up to float32), (5) Output file FX tail duration in milliseconds, (6) roundtrip audio latency compensation in milliseconds, and (7) which audio interface output and input channel(s) to use (i.e. which channel is physically routed to the appropriate analog device). Given the successful reception of each of the above parameters, max-io loads the input file into RAM, and allocates the appropriate memory (given the FX tail and sample format) to record the return output file. When all is set, max-io plays back the specificed digital audio routed to the appropriate [dac] channel, while simlutaneously starting recording on the appropriate incoming [adc] channel. Neither the amplitude of the outgoing digital signal, nor the amplitude of the incoming analog signal is adjusted. When playback is complete, the audio buffer containing the analog-effected audio is trimmed, given the latency compensation parameter, and the file is writen to disk with correct specified sample format.

Furthermore, given the specifics of the system– the software and hardware resources of the PC and the audio interface hardware connected to it–the MDC can adjust DSP parameters for each job by interfacing with max-io. For instance, different interfaces may be selected, along with different signal vector sizes and sampling rates. This flexibility and customizability built into the ground floor of the RackFX system makes it possible to potentially run this automated system on a variety of machines with different limitations interfacing with different audio gear.

5. FUTURE DEVELOPMENTS

Future development using the RackFX platform is focused on not only extending the device offerings for the analog processing of any given job, but also providing users with the ability to preview the analog audio effect, ideally by routing a small portion of audio through the device to test the current parameter settings before processing the whole file. User ability to interact with the web GUI such that they may turn the appropriate virtual knobs and preview the effects of different parameter settings is highly desirable and would make the platform even more useful to the non-expert engineer or musician looking to experience the possibilities of analog processing.

Furthermore, while the on-site facilities supporting the RackFX platform are steadily growing in the number of available analog devices, it is also possible for the RackFX platform to be backed by a distributed network of device providers– partners existing in multiple physical locations sharing their own analog devices, making their devices available to internet users through the RackFX web-app. The notion of scalability here is particularly interesting and encouraging because once affiliates are provided with the necessary software (MDC + max-io) and the robotics hardware to mount onto their particular analog device(s), the RackFX platform could grow to allow individuals and professional studios alike to share their analog hardware resources.

6. CONCLUSIONS

The RackFX platform allows users to access analog equipment through an easy to use web site. This platform allows users to use audio processing equipment through cloud-based technology and robotics. And in the future, studios and individuals can bring their devices to the community and become a RackFX partner, bringing analog processing capability to users around the world through our easy-to-use custom framework.

Ultimately, RackFX represents an opportunity for musicians and audio producers to engage in the analog processing of sound through the web. In the past, low-budget musicians, video producers, music producers and podcasters had to rely on increasingly expensive digital plugins that attempt to emulate analog signal processing devices, or they had to invest in cheap analog gear with low-quality components in an attempt to achieve the sound qualities they associate with high-budget studio analog gear. Now users can have access to this high-end equipment through the RackFX platform.

As a digital music solution, the RackFX project simply refuses to pick sides in the analog versus digital signal processing debate. While our commitment to achieving ever more refined in-the-box DSP techniques and analog device emulations will continue, we should not be dogmatic here; we should not think that *parity* between the digital and analog world is either necessary or desireable. Nor should we eschew what digital tools have afforded in the name of maintaining limited access to analog processing units– resulting in analog fetishization to an even greater degree, given such a scarce resource. By leveraging a host of digital technologies, including cloud computing, real-time digital audio manipulation, and robotics, the RackFX platform provides an alternative path: make analog devices accessible through the web to empower all musicians, regardless of budget. At the very least, our psychoacoustic value judgements regarding the 'warmth' and 'presence' of analog processing effects will be put to the test now that analog gear is no longer cloistered. Ideally, a platform like RackFX will help advance our ability to hear.

7. REFERENCES

[1] M. Karjalainen and J. Pakarinen, "Wave Digital Simulation of a Vacuum-Tube Amplifier," *ICASSP*, vol. 2, pp. 153–156, 2006.

[2] "Waves Audio Analog Models Plugins," http://www.waves.com/plugins/analog-models, accessed: 2-29-2016.

[3] J. O. Smith III, "Physical Modeling Synthesis Update," *Computer Music Journal*, vol. 20, no. 2, pp. 44–56, 1996.

[4] "Silophone project," http://www.silophone.net/eng/about/desc.html, accessed: 4-20-2016.

[5] "Patchwork: Control a Massive Modular Synthesizer," http://synth.media.mit.edu/patchwerk/, accessed: 4-20-2016.

Composing and Performing Digital Voice Using Microphone-Centric Gesture and Control Data

Kristina Warren
University of Virginia
kmw4px@virginia.edu

ABSTRACT

Digital voice is a rich area for compositional and performative research. Existing voice-technology work frequently entails a division of labor between the composer-technologist, who creates the hardware/software, devises the formal structure, and writes about the work; and the performing vocalist, who may have some creative or improvisational input. Thus, many scientific papers on digital voice lack an authorial performance perspective. This paper aims to imbue performance back into the discussion of digital voice, drawing from my own experience as a composer-technologist-vocalist. In addition, many novel controllers for vocal performance are glove- and hand-centric, but in fact the hands are auxiliary to vocal performance. I propose the Abacus interface, which is mounted directly on the microphone, as a more literally voice- and mouth-centric means of controlling digital voice. The Abacus treats rhythm, pitch, and noise parametrically and tracks vocal gesture input to modulate among various processing states.

1. INTRODUCTION

Digital voice comprises a vast, rich sonic palette. Though we are accustomed to considering the abstract compositional voice, and to using voice as an inspiration for developing affective audience connection in composition [1], the area of digital voice demands more focused research. Few authors of scientific papers on digital vocal compositions are themselves vocalists, and many voice-based controllers are rooted in hand motion and have very little to do with the precise gestural work possible in the lips, teeth, tongue, and vocal tract. I propose that the mouth is a prime site of vocal control, and thus my mic-mounted interface called the Abacus takes steps toward evaluating the gestural and control potential of digital voice.

2. RELATED WORK

Many early voice-technology works were tape compositions whose primary source material was recorded speech. Such speech-based compositions still thrive, particularly in the legacy of Swedish text-sound composition and related compositional styles [4, 9, 16]; analysis of these works emphasizes intelligibility of the text [2].

More recent voice-technology works, by contrast, explore timbre and the act of performance. Works by Trevor Wishart exemplify this compositional style. Wishart typically records a vast array of sounds, many based in extended vocal techniques, and subsequently applies frequency-domain transformations, such as spectral shifting and stretching, to create fixed media compositions [15].

Today, works for digital voice emphasize live vocal processing – particularly video [10] and polyphonic [12] extensions of voice – and hand- and limb-centric controllers. The visual beauty of glove- and hand-based controllers, such as those employed by Imogen Heap, Laetitia Sonami, and Pamela Z, guides the audience toward an understanding not only of digitized vocal sound but also of nuances of performer affect and identity [7, 11, 18]. Nonetheless, despite the primacy of the hands in the realm of haptic sensing and gesture [14] the hands are in fact auxiliary to vocal performance, and therefore more study is needed of voice- and mouth-centric controllers. Furthermore, the writing on digital vocal music is dominated by the perspectives of non-vocalist composers (notable exceptions include [6, 13, 17]), so it is necessary to rediscover the performance perspective within digital vocal music.

3. DIGITAL VOICE

3.1 Live Processing in Max/MSP

In performance, I employ two Max/MSP units I built for live vocal processing: [rhyGran] and [glchVox]. These consist of rhythmic, granulation, and frequency-domain effects of my own making, as well as Michael Norris' Soundmagic Audio Unit plug-ins for spectral processing.[1] Both units can use either live or prerecorded samples; [rhyGran] tends to produce normative, voice-like sounds, while [glchVox] often yields prominently digital, glitchy sounds, but both modules can be variably employed. Processing on the live voice signal is similar, alternately maintaining or ablating the vocal character of the original signal. This patch is meant to work in conjunction with my vocal performance practice, which consists of both extended techniques and more traditional singing styles.

3.2 Performative Affect, Novel Controller

My recent research has turned increasingly toward the matter of performative affect. I find that the potency of mouth as source of gestural and control data is enhanced by eye contact and engagement with the audience. Thus, the Abacus is part of an effort to streamline my own performance affect.

In early work with the aforementioned Max/MSP modules, I used a tablet (first a Wacom tablet[2] and subsequently a Google Nexus tablet[3]) to control live processing of the voice. Later, I decided for aesthetic and performative reasons that the tablet was not sufficient as a controller. First, the sonic output of my work tends to be dense and sculptural, and the two-dimensional tablet feels unsuited to this sound. Moreover, I experienced a growing desire to get "out from behind" the laptop in order to develop a greater connection with the audience. My work points toward the mouth as a crucial site of voice. While hands and limbs aid in expressivity, they are in fact auxiliary to vocal performance, so I began to develop a novel controller to be mounted directly on the microphone . I argue that the most important vocal gestures originate in the mouth and vocal tract, rather than the hands.

4. ABACUS INTERFACE

4.1 Versions 1 and 2

The first two versions of the Abacus used an Arduino Teensy[4] to send digital and analog control input to the central Max/MSP patch, and included several LEDs to provide visual feedback during performance. These early Abacus versions were mounted on a breadboard/protoboard.

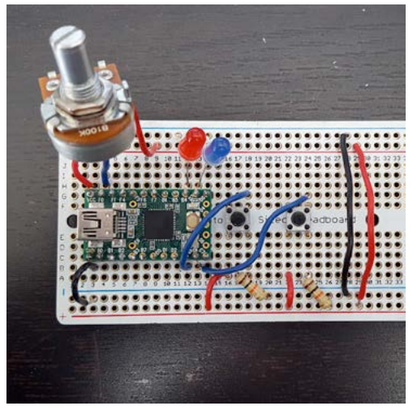


Figure 2. Abacus version 1.

4.2 Versions 3.0 and 3.1

Abacus 3.0 and 3.1 continue to use an Arduino Teensy and maintain the exposed-wires aesthetic of the earlier versions. Version 3.1 dispenses with the breadboard and instead uses thermoplastic and suede cabling to achieve direct adherence to the microphone and clip. Abacus 3.1 consists of 8 toggles, 1 button, 2 LEDs, and 1 potentiometer. The toggles are the primary source of control data; the button triggers recording of live vocal samples; the LEDs report state and time information, and the potentiometer controls gain.

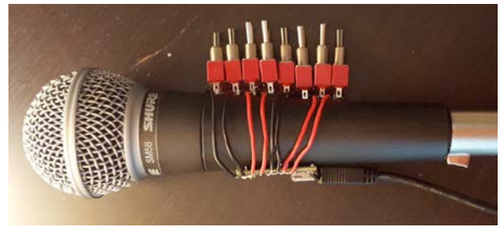


Figure 1. Abacus version 3.0.

Inspired by the ancient adding tool of the same name, the Abacus treats control data parametrically. There are three possible control states: Rhythm, Noise, and Pitch. Each control state consists of a three-dimensional control space whose axes are a Short Term parameter, a Long Term parameter, and a Texture parameter. Toggles 3-4 allow navigation along the x-axis, Short Term; Toggles 5-6 along the y-axis, Long Term; and Toggles 7-8 along the z-axis, Texture. Each pair of Toggles (3-4, 5-6, 7-8)

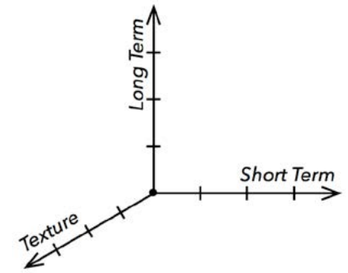


Figure 3. 3D control space, Toggles 3-8.

outputs four possible values: 0, 1, 2, 3. Thus, there are 4^3 = 64 possible positions within this 3D control space.

Toggle #	Function
1	Backward/forward among states (saved before/during performance)
2	Listen to RVGs
3-4	Short Term control axis (x)
5-6	Long Term control axis (y)
7-8	Texture control axis (z)

Table 1. Toggle functions.

Routing Vocal Gestures (RVGs) determine the control state (Rhythm, Noise, or Pitch); examples of Routing Vocal Gestures are shown in Table 2. There are two important limitations on the control ability of the RVGs:

(1) **RVGs do not directly control sound processing**. Instead, they route control data from Toggles 3-8.
(2) **Toggle 2 controls whether the patch listens for RVGs.** If Toggle 2 is in the "off" position, an RVG will not cause a change in control state.

[1] http://www.michaelnorris.info/software/soundmagic-spectral

[2] http://www.wacom.com/en-us
[3] https://www.google.com/nexus/
[4] https://www.arduino.cc/

My habit is to vocalize continually throughout a given piece, establishing a symbiosis between the sounds emerging directly from my mouth and those emerging from the Max/MSP patch. Because of this continuous vocalization, it is disadvantageous for RVGs to be too sensitive. Thus, though RVGs can serve a control function, they are intended to be primarily gestural.

Example Routing Vocal Gesture (RVG)	**Resulting Control State**
"inhale k" = unpitched; inhale with occasional tightening of soft palate, yielding sucking "k" sound	Rhythm
"hum n" = pitched; hum, rapidly touch tongue to front palate, yielding closed-mouth "n" sounds	
"8vb" = pitched; harmonic undertone from false vocal fold vibration at slower frequency (e.g. f/2)	Noise
"fast in/ex" = unpitched; rapid inhales, exhales; hyperventilation-like	
"ae" = pitched; short nasalized repetitions of [ae] vowel (as in "cat")	Pitch
"lip squeak" = pitched; upper teeth on moistened lower lip, inhale yielding one or more gliss pitches	

Table 2. Example RVGs and control states.

Control x-axis (Short Term) = Toggles 3-4	**Control y-axis (Long Term) = Toggles 5-6**	**Control z-axis (Texture) = Toggles 7-8**
Control State = Rhythm Triggered by Rhythm RVG, e.g. "inhale k"		
Meter 0 = slow, irregular … 3 = fast, regular	Loop 0 = 10-20% chance repeat same rhythm … 3 = 80-90% chance	Sync 0 = 10-20% slaving to master rhythm … 3 = 80-90%
Control State = Noise Triggered by Noise RVG, e.g. "8vb"		
Timbre 0 = dark … 3 = bright	Cyber 0 = mostly unaltered voice … 3 = mostly processed	Density 0 = indiv. lines discernible … 3 = wash of sound
Control State = Pitch Triggered by Pitch RVG, e.g. "ae"		
Interval 0 = mostly small (< min3) … 3 = mostly large (> Maj6)	Continuity 0 = scattered, granular … 3 = phrasing, key apparent	Solo 0 = accomp. live voice … 3 = soloistic counterpoint

Table 3. Control states and values.

The three axes of the 3D control space are Short Term, Long Term, and Texture. This organization: (1) allows both detailed and general control during performance, (2) promotes a balance between composition and improvisation, and (3) acts as a mnemonic during performance. It is somewhat conceptually difficult to establish an exact parallel between, for instance, the Short Term axis in the Rhythm and Pitch control states, or between the Long Term axis in the Rhythm and Noise control states. Nonetheless, the Short Term axis is meant to give some information about the immediate character of the sound; the Long Term axis, about the phrase-level organization of sounds; and the Texture axis, about the relationship of patch voices or layers to one another.

Several control axes bear further explanation. "Cyber," the Long Term axis of the Noise control state, refers to the frequency of timbre changes and thus the implied "cyberization" of vocal samples. "Continuity," the Long Term axis of the Pitch Control state, refers to consonance and apparent modulation. "Solo," the Texture axis of the Pitch control state, describes the extent to which the digital voices or layers form a singular, coherent counterpoint to my vocal input.

Finally, Toggle 1 allows toggling backward to the previous state. If I arrive at a desirable configuration of settings, I can save this and return to it later using Toggle 1.

5. IMPLEMENTATION OF ABACUS

5.1 *couldn't*, voice and stereo audio

My piece *couldn't* (2016)[5] is a studio-based composition made with Abacus 3.1; it comprises two sections. The first section consists mainly of rhythmic manipulations of the live voice signal and some live samples thereof. The second section emphasizes noise and textural density; I recorded my vocal performance with several microphones and fed these dry and processed versions back into the Abacus as gestural data during mixing. I made a compositional decision to use "inhale k," "8vb," and "ae" as the RVGs for Rhythm, Noise, and Pitch respectively; these techniques as well as proximity work among the microphones comprise the bulk of the vocal performance.

A primary goal of *couldn't* is to use the Abacus to blend compositional and improvisational work. The main vocal line was a single improvised performance in the studio, eight minutes in duration. This was recorded with the Shure SM58 microphone on which the Abacus is mounted, and simultaneously with a Rode NT1-A. This Rode track was then fed back into the Abacus during mixing to add layering and depth. The Rode's greater sensitivity yielded a more precise translation of vocal signal into control data. I played with temporal displacement, sometimes sending control information from the Abacus at the same time points in the fixed Rode track as I originally did when performing with the Shure, but at other times developing a new temporal control path. All together, these layers give a sense of ghosts in the performance: the live voice and its associated processing do not always co-occur.

[5] Audio available at nitious.bandcamp.com

6. FUTURE DIRECTIONS

I intend to incorporate text into the interface design. I am interested in text as both sonic/semantic material and instruction for action. Thus, incorporating a live text video score will shape my body and mouth in real time, and will allow the audience a stylized close view of my vocal work.

The voice inherently carries much timbral and stylistic flexibility, and recent spectral analysis of recordings [8] and laryngoscopic studies of expressive and extended vocal techniques [3] begin to shed light on the vast performative potential of voice. More study is needed to integrate this directly with the practice of technologized academic music. In addition, singing voice synthesis [5] is a vast and promising field which demands greater connection to digital-vocal work, for instance as a performance partner to a human vocalist.

Acknowledgments

I am grateful to Peter Bussigel for his guidance in designing and building the Abacus interface, and to Daniel Jolliffe, Collin Bradford, Thomas Ouellet Fredericks, and Seejay James for their development of Serial-based Arduino and Max patches that allow communication between the Teensy and Max/MSP.

7. REFERENCES

[1] C. Bahn, T. Hahn, and D. Trueman, "Physicality and Feedback: A Focus on the Body in the Performance of Electronic Music," *Proceedings of the 2001 International Computer Music Conference*, Havana, 2001, pp. 44-51.

[2] A. Bergsland, "The Maximal-Minimal Model: A framework for evaluating and comparing experience of voice in electroacoustic music," *Organised Sound*, vol. 18, no. 2, pp. 218-228, 2013.

[3] D.Z. Borch, J. Sundberg, P.-Å. Lindestad, and M. Thalén, "Vocal fold vibration and voice source aperiodicity in 'dist' tones: a study of a timbral ornament in rock singing," *Logopedics Phoniatrics Vocology*, vol. 29, no. 4, pp. 147-153, 2004.

[4] W. Brunson, "Text-Sound Composition – The Second Generation," *Proceedings of the Electroacoustic Music Studies Network 09*, Buenos Aires, 2009.

[5] P. Cook, "Singing Voice Synthesis: History, Current Work, and Future Directions," *Computer Music Journal*, vol. 20, no. 3, pp. 38-46, 1996.

[6] M. Guilleray, "Towards a fluent electronic counterpart of the voice," Masters thesis, Institute of Sonology, Royal Conservatory of the Hague, 2012.

[7] D. Haraway, *A cyborg manifesto: Science, technology, and socialist-feminism in the late 20th century*, Springer Netherlands, 2006.

[8] S. Lacasse and C. Lefrançois, "Integrating Speech, Music, and Sound: Paralinguistic Qualifiers in Popular Music Singing," *Proceedings of 2008 IRCAM Expressivity in Music and Speech Conference*, Campinas Brazil, 2008.

[9] C. Lane, "Voices from the Past: compositional approaches to using recorded speech," *Organised Sound*, vol. 11, no. 1, pp. 3-11, 2006.

[10] G. Levin and Z. Lieberman, "In-Situ Speech Visualization in Real-Time Interactive Installation and Performance," *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering*, Annecy France, 2004.

[11] G. Lewis, "The Virtual Discourses of Pamela Z," *J. Society for American Music*, vol. 1, no. 1, pp. 57-77, 2007.

[12] P.J. Maes, M. Leman, K. Kochman, M. Lesaffre and M. Demey, "The 'One-Person Choir': A Multidisciplinary Approach to the Development of an Embodied Human-Computer Interface," *Computer Music Journal*, vol. 35, no. 2, pp. 22-35, 2011.

[13] C. Stamper, "Our Bodies, Ourselves, Our Sound Producing Circuits: Feminist Musicology, Access, and Electronic Instrument Design Practice," Masters thesis, Mills College, 2015.

[14] C. Udell and J. Sain, "eMersion | Sensor-controlled Electronic Music Modules & Digital Data Workstation," *Proceedings of the International Conference on New Interfaces for Musical Expression*, London, 2014, pp. 130-133.

[15] T. Wishart, "The Composition of 'Vox-5,'" *Computer Music Journal*, vol. 12, no. 4, pp. 21-27, 1988.

[16] A. Woloshyn, "The Recorded Voice and the Mediated Body in Contemporary Canadian Electroacoustic Music," PhD dissertation, University of Toronto, 2012.

[17] A. Young, "The Voice-Index and Digital Voice Interface," *Leonardo Music Journal*, vol. 24, pp. 3-5, 2014.

[18] M. Young, "Latent body – plastic, malleable, inscribed: The human voice, the body and the sound of its transformation through technology," *Contemporary Music Review*, vol. 25, no. 1-2, pp. 81-92, 2006.

Composing for an Orchestra of Sonic Objects: The Shake-ousmonium Project

Otso Lähdeoja
University of the Arts
Sibelius Academy
PL 30, 00097 Taideyliopisto
Helsinki, Finland
otso.lahdeoja@uniarts.fi

ABSTRACT

This article reports and discusses the "Shake-ousmonium" project; a collective effort to design and build an orchestra of sonic objects, in parallel to the composition and performance of five original pieces in a concert. A significant diversity of sound sources were created using structure-borne sound drivers to transform a range of materials into loudspeakers, as well as augmented instruments, DIY electromechanical instruments and prepared speakers. The article presents the system design and the pieces composed for it, followed by a discussion on the extension of the compositional gesture towards the material environment – the sonic objects. Audiotactility in concert setting is considered, in connection with the results of an audience feedback poll conducted after the concert.

1. INTRODUCTION

This article presents and discusses the "Shake-ousmonium" project, developed at the Sibelius Academy Centre for Music and Technology during the autumn 2015 and culminating with a concert on November 19, 2015. The shake-ousmonium constituted a project combining artistic research with technological development, namely composition with experimental sound diffusion techniques. The name refers to the established Acousmonium tradition of loudspeaker orchestras in electroacoustic music. Over the years, the electroacoustic diffusion practice has given rise to state-of-the-art loudspeaker ensembles combining speakers of different sizes, sonic characteristics, and radiation patterns. Notable and documented Acousmoniums include the INA-GRM Acousmonium at Radio France [1], The ZKM Klangdom in Karlsruhe [2], the Huddersfield Immersive Sound System [3], and the Birmingham Electroacoustic Sound Theatre [4].

In reference to the Acousmonium, the Shake-ousmonium explores the artistic possibilities emerging from an orchestra of sonic objects. The project winks at the Acousmonium by bringing onstage a bestiary of miscellaneous sound objects: vibrators, tactile transducers, motors, prepared instruments and loudspeakers, vibrating seating, paper, metal and plastic. General purpose PA/hi-fi sound is replaced by composed objects, extending the gesture of composition towards the material environment.

The Shake-ousmonium stems from an ongoing research project on structure-borne sound in music and intermedia creation which aims to explore and enact the artistic potential of audio-rate vibration driven into solids and objects turned into loudspeakers. The rationale of our project is at the same time technological and musical, rooted in the research-creation methodology where artwork and technological development are brought into mutually nourishing dynamics [5].

The Shake-ousmonium was designed as a collective effort to build an orchestra of sonic objects, compose music for it and perform the pieces at the Sibelius Academy's annual MuTeFest at the Helsinki Music Centre. Five pieces with very different approaches and aesthetics were completed and performed at the final concert, authored by Andrew Bentley, Kalev Tiits, Alejandro Olarte, Andrea Mancianti and Otso Lähdeoja.

This article presents the system design and set-up in connection with the compositions, followed by a discussion on the interest of alternative audio diffusion techniques and compositional strategies, as well as the related aesthetic choices. Finally, a case study on the use of audiotactility in concert setting is presented with the results of an audience feedback poll.

2. BACKGROUND

The idea of a sonic object orchestra was pioneered by David Tudor and implementd in the different iterations of his piece "Rainforest": "My piece, "Rainforest IV", was developed from ideas I had as early as 1965. The basic notion, which is a technical one, was the idea that the loudspeaker should have a voice which was unique and not just an instrument of reproduction, but as an instrument unto itself" [6].

The "Rainforest" is a concert-installation, a collection of sculptural objects with surface transducers and piezo microphones, distributed in a space and performed live. Audio signals are driven into the objects, making them vibrate and emit sound. The sound radiating from the objects is picked up by piezo microphones and amplifies via a regular loudspeaker system. Tudor's piece states that each object has it's own sound source and emphasizes that the object's construction and performance are interconnected.

The Shake-ousmonium project develops a different perspective from Tudor's seminal idea. Instead of having a tight object-sound coupling and a collective performance-happening, we decided to explore the object-orchestra as a medium for the composition and diffusion of different individual compositions, following the Acousmatic tradition. Each participant of our team composed a piece exploring a personal interpretation of the general idea. However, unlike the Acousmonium, the Shake-ousmonium project gave rise to pieces with a strong performative aspect. Each piece was performed live and the stage presence constituted a central part of the pieces.

In his theoretical works, Horacio Vaggione develops the notion of a "composable space" (*espace composable*), describing compositional processes as weaving relations between sonic and virtual objects, themselves composed at different levels of detail [7]. In the Shake-ousmonium case, Vaggione's *objet composable* is expanded towards the material realm. The physical sonic objects are themselves crafted as inherent parts of the composition, along with the notational, digital or gestural entities.

Some aspects of the Shake-ousmonium evoke Agostino di Scipio's ecosystemic approach to signal processing and composition [8]. The homology is particularly present in Alejandro Olarte's and Andrea Mancianti's pieces (see section 4.), where Di Scipio's interrelation between man, ambience and machine are transposed to man, object and machine. Other related works include Pierre Bastien's mechanical instrument-automatons [9], as well as Lynn Pook and Julien Clauss' performative installation "Stimuline" where audiotactile vibration is used in a musical context [10]. The effect of audiotactile vibration in music listening has been researched by Merchel and Altinsoy, concluding that the listening experience was enhanced by the addition of the haptic channel [11].

3. SYSTEM SET-UP

The Shakeousmonium was designed and implemented over a six-month period onwards from June 2015. The concert took place at the Helsinki Music Centre's Black Box, a 30 x 15 x 8m venue intended for electronic and amplified music performance. The object orchestra was placed at the center of the space, with the public seated on 70 chairs mounted on three "islands" of stage risers, equipped with bass-range structure-borne audio drivers. The setup gave an installation-like impression with all its diverse DIY curiosities. With the exception of a double bass, the whole concert featured only self-constructed, modified or prepared objects and instruments. The installation gave rise to a multidimensional and multimodal concert experience: air-borne sound was radiating from objects on floor level as well as from suspended elements above. Audiotactile vibration was driven into the audience's seats, enabling for a haptic perception of the bass frequencies. The sculptural sonic objects and self-made instruments added a strong visual element to the show.

Summing up the different sonic objects used in the concert gives the following list: Audio signals were driven into plastic panels, metal sheets, stage risers with audience seating on top, bass drum and electric guitars, all equipped with structure-borne sound drivers. Electromechanical instrument included solenoids and electric motors activating diverse sound-making mechanisms on metal, plastic, wood and even a ship in a bottle. Traditional loudspeakers were prepared using paper and plastic.

Figure 1. Setting up the Shake-ousmonium. The image shows a selection of sonic objects, augmented instruments and electromechanical instruments constructed for the concert.

4. THE SHAKE-OUSMONIUM PIECES

Five pieces were created and performed for the Shake-ousmonium project, all in which the compositional gesture included the musical material, software implementation as well as the design, construction and spatial distribution of the objects used for diffusion.

Alejandro Olarte's "Hephaestus Song" (2015) opened the concert. The piece is built around a large suspended metal sheet, equipped with a pickup, transducers and a feedback system regulated with dsp operators and audio descriptors. The piece is a study of the potentialities of one material to be simultaneously the control interface, the sound exciter and the sound source in an electroacoustic instrument. The composition voluntarily restricts itself into the boundaries of the sole metal sheet, exploiting the whole extent of its sonic possibilities in reference to Hapheastus, the blacksmith god of Olympus.

Andrea Mancianti's "Preparatory Studies for Controlled Autophagia" (2015) stages a bass drum and two electric guitars mounted with vibration speakers. The performer is equipped with a self-built glove-microphone using a low-cost physiotherapy palm support. The piece explores the possibility of using the resonating behavior of an object set inside a feedback loop, and at the same time to find strategies to perform and improvise with it. With the hand-held microphone, the performer is able to "ignite" and manipulate the feedback, allowing for an intuitive explorative performativity. A Max/MSP patch controls the feedback levels and

adds routes the audio through a chain of effects, replicating and extending the principles of acoustic feedback-through-a-medium in the digital domain.

Otso Lähdeoja's "Tapage Nocturne for Double Bass, Video and Electronics" (2015) is a mixed music piece for live bass player and projected video replicas of the performer. The piece's sound diffusion system comprises bass frequencies driven into the audience seating, an array of plexiglass panel speakers as well as traditional cone speakers prepared with paper for buzz-like resonances. Four video avatars of the player are projected on a screen, engaging a play of relations with the live instrumentalist. The composition is based on a deconstruction of traditional double bass roles and models and it is conceived as a detailed study of basic gestures available on the bass: hitting, plucking, rubbing and bowing the strings as well as the instrument's body.

Andrew Bentley's "Improvisation for Shake-ousmonium Instruments" (2015) brought onstage a quatuor of hand-crafted sound machines: the "Diapason", "Sheet Shaker", "Ping Pong Shaker" and "Ships". The instruments use motors, solenoids and loudspeakers to produce a variety of sounds from mechanical noise to altered reproduction of audio signals. The instruments are controlled via max/MSP and Arduino boards with sub-audio or audio-rate signals. The instruments were performed in live as an electroacoustic duo improvisation.

Kalev Tiits' "Music Without Computers" (2015) presents an orchestra of electromechanical devices driven by motors and solenoids. The piece's structure emerges from processes that run on DIY logic circuits built from discrete transistors – primitive digital logic, not following Von Neumann architecture. The parts bin used in the piece contains objects taken from various sources, including washing machines, motorcycles and bicycles, added with bits fabricated especially for the piece.

Figure 2. Nathan Thomson performing "Tapage Nocturne" for Double Bass, Video and Electronics. The piece incorporates plexiglass panel speakers, prepared loudspeakers as well as an audiotactile public address system.

Figure 3. Alejandro Olarte performing "Haphaestu's Song", a composed sonic ecosystem comprising metal plates, transducers, contact microphones and digital signal processing.

5. COMPOSITIONAL STRATEGIES FOR SONIC OBJECTS

The central question regarding the use of sonic objects in an electronic music context pertains to articulating the interest of alternative diffusion techniques as opposed to traditional loudspeakers. The traditional cone speaker is the universal sound actuator in the present cultural context. Its technology has been perfected over a century, resulting in spectacular refinement in spectral, spatial and dynamic reproduction of sound. The loudspeaker is able to offer a quasi-transparent medium for actuating sounds. The ideal of a perfect reproduction aims towards the disappearance of the speaker-interface altogether. A perfect speaker would not translate a given signal into sound waves, it would flawlessly transduce the signal in every detail, thus *being* the sound itself. The loudspeaker is so universal that it has blended into being an inherent part of our hearing culture, somehow becoming physically transparent as well. When listening through speakers, one often focuses on the sound itself, discarding the interface. The speaker's function is precisely that: allow for the listener to reach out to a purely sonic realm by fading away the transmitting medium. At the same time, and in parallel to its universality, the cone speaker is also an object with defined characteristics such as radiation pattern, frequency and dynamic response, as well as material and visual attributes.

One might argue that the aural percept given by a loudspeaker is very different from an acoustic instrument. The perceived spatial diffusion, dynamics, presence and timbre are distinct and immediately perceptible for each case. In our research framework, we have adopted the term "aural imprint" to signify the perceptual attributes of a sound source. In the case of a cone speaker, its aural imprint is characterized by a conical radiation pattern and the related spatial reflections, the capacity to channel significant amounts of acoustic energy into the sound "beam", as well as the individual frequency and dynamic properties of each speaker model. The aural imprint of, for example, a harpsichord, is completely different. The horizontal soundboard radiates energy primarily on a vertical axis, towards the ceiling and the floor instead of towards the public, resulting in a more spatially immersed aural image. The tone woods and design create a specific timbre for each instrument.

Approaching audio diffusion via sonic objects such as those used in our project is a deliberate deviation from the ideal of transparent reproduction represented by the cone speaker. The sonic objects act as material filters, affecting the audio reproduction by their own physical properties such as resonant modes. The dynamic and frequency ranges can be severely restricted. Moreover, physical noise and distortion resulting from material vibrations may occur, especially at higher gain levels.

The interest of alternative audio diffusion techniques, such as the sonic objects used in the Shake-ousmonium project can be articulated via the "aural imprint" concept. By designing a sonic object, one is designing its aural imprint, or sonic signature; the way it translates audio signals and radiated them into a space. The design gesture is inherently related to the compositional process itself, as the aural imprint is an important factor in the esthesis as well as the aesthetics of the piece. Unlike the loudspeaker, here the actuator becomes a distinctive sonic object with its aural imprint crfted as a part of the overall artistic gesture. The filtering effect, frequency range, radiation pattern, spatial localization and material resonances are brought within the compositional process, offering a wide terrain for experimentation and innovation.

Moreover, sonic objects have a distinct physicality and appearance, which likewise become parameters for composition. Composing for an array of glass panel speakers, metal sheets or active acoustic instruments has not only sonic, but also visual and scenographic implications. The superposition of a material object and sound holds a vast potential for constructing poetics and meanings. Some other strategies of "sonic scenography" have been sketched in a previous publication [12]. Another enchanting perspective is the audiotactile channel. Surface vibrators enable for the sound to become into direct physical contact with the listener, offering the modalities of that contact as additional material for the composer (see section 6 for further development on the use of audiotactility in our project).

The pieces composed for the shake-ousmonium tapped into this expanded compositional terrain via different strategies. An analysis of the pieces brings up the following set of approaches:

1) Sound spatialisation via an array of sonic objects. The objects' aural characteristics and spatial localization are an inherent part of the composition.

2) Superposition of aural and visual elements. The sculpturality of sonic objects becomes part of the composition and engages a dialog with the sound.

3) Instrument as a speaker. Audio driven into traditional instruments creates a double-layered sound source: electronic sounds can be diffused in parallel to the instrumental performance.

4) Sonic ecosystem embedded into the materiality of the sounding objects. Signals are driven into solids and picked up by contact microphones, giving rise to a self-maintaining loop comprising material, analog and digital elements, all of which can be included into the compositional process.

In summary, composing for sonic objects presents two sides: on one it restricts the fidelity of audio diffusion by a non-neutral interface, on the other it offers possibilities for artistic strategies and poetics by expanding the compositional gesture towards the material environment. Sonic objects employed in a concert setting give rise to a singular space in-between the categories of installation art, live concert and the Acousmonium.

6. AUDIOTACTILITY IN CONCERT SETTING

The Shake-ousmonium setup comprised a system for driving bass frequencies into the audience's seats, giving rise to audiotactile perception and opening the possibility to extend the compositional gesture towards the tactile perceptual channel.

The system was implemented by mounting the audience seats on stage risers and attaching a low-frequency sound driver (Visaton BS 130-4) to each element. Twelve drivers were used to cover 70 seats, dispatched from a mono source. The drivers were powerful enough to give a distinct sensation of vibration, texture and attacks on the lower part of the body, especially on the soles of the feet. However, the system did not provide cinema-type special effects shaking the whole body.

Figure 4. Audiotactile vibration was driven into stage risers under the audience seating, enabling to activate the haptic channel as a compositional parameter.

The "audiotactile public address" system was used in Otso Lähdeoja's piece "Tapage Nocturne" as a prolongation of the double bass via two distinct compositional strategies. Firstly, being a mixed piece, the live bass playing suffered initially from a perceptual parity with the pre-recorded sources. Driving the bass to the audience seating dramatically enhanced the perceptual presence of the instrument, giving rise to a perceptual zoom effect; the bass seemed to be nearer, in direct contact with the listener and clearly in relief in relation to the solely air-borne diffused sounds. Secondly, the system's audiotactile capacity was used as a compositional element for creating narrativity and dramaturgy within the piece. Audiotactile intensity was at some parts coupled with bass playing intensity, at other moments its presence/non presence was composed as an internal "respiration" of the piece. Depending on the audiotactile presence, the piece could be made feel intense, close, and charged, or on the contrary, aerial and distant.

6.1 Audiotactility – Audience Feedback

The piece being part of a research project on audiotactility in concert setting, we polled a selected group of audience members via email after the concert with the following question: "The shakeousmonium involved vibrating stage risers for the audience seating, driving bass vibrations into the soles/bodies of the public. How would you qualify your experience of audiotactility in a concert setting?"

We received nine responses out of twenty email queries sent (the total audience for the concert was 70), four from non-specialists and five from people engaged in electronic music practice. The audience feedback was overall enthusiastic about the introduction of a tactile dimension into a live concert experience. All respondents agreed that audiotactility added something to the reception of the piece, and did not hinder or counter the musical experience.

More analytical reflections were also received, giving valuable and detailed feedback about the audiotactile experience. One thread of comments emphasized the difference between a traditional sub-bass woofer and an audiotactile bass system. The respondents suggested that an audiotactile system enables to achieve dramatic low frequency percepts with low decibel levels. In order to have a physical effect – to "touch" the listener - a sub-woofer has to be operated on relatively high gain levels, leading to the impressive PA systems used in bass-emphasizing genres like beat-based electronic music. With an audiotactile system, it is possible to "touch" the audience within chamber-music like sound levels.

Also, a difference of corporeal reception was noted. In our system, the bass vibrations were perceived through the feet and lower abdomen, whereas the sub-bass woofer is more perceived in the chest, giving rise to two distinct sensations. This finding points towards the possibility to combine both systems into a compositional strategy: different bass techniques and related corporeal percepts can be used as material for composition. One responded suggested that it would be interesting to run comparative tests between sub-woofer and audiotactile bass systems in a laboratory setting.

The specialist respondents agreed that sound source perception was enhanced by the audiotactile system. The live bass playing conducted to the audience seats felt close, precise and clear, as opposed to the air-borne sound diffusion. There was no feeling of latency or perceptual gap between the double bass's acoustic sound and the tactile vibrations. Another aspect mentioned was the perceptual familiarity of audiotactility. At the era of omnipresent loudspeakers, virtually everyone has experienced vibrating surfaces and audio-rate vibration infusing the body, most commonly at the cinema and in dance music clubs. However, the potential of the technique for concert music was appreciated, enabling a multitude of compositional possibilities.

7. CONCLUSIONS AND FUTURE WORK

The Shake-ousmonium project was built on a collective synergy of five composer-researchers as a one-of-a-kind concert. There is no plan at the moment to rebuild the system in all of its detail and diversity. However, the musical and experiential outcomes of the project are finding their ways into current and future projects. Most substantially, we have founded a regular ensemble experimenting with the possibilities of alternative speakers, sonic objects and scenographies as well as audiotactility. The project is entitled "Electronic Chamber Music", and it is designed to function like a "band", forging an original repertoire and giving regular concerts. Also, an experimental loudspeaker workshop is being taught at the Sibelius Academy by Prof. Andrew Bentley, giving rise to a new generation of sonic objects that could form a future iteration of the Shake-ousmonium concept.

Acknowledgments

The author would like to express their gratitude towards the Academy of Finland for funding the present research project on structure-borne sound, as well as the Sibelius Academy Music Technology Department and MuTeFest for producing the Shake-ousmonium concert. Credits for the photos used in this article: Antti Ahonen.

Figure 5. The Shake-ousmonium in action. A "tutti" improvisation was performed at the end of the concert.

8. REFERENCES

[1] INA-GRM Acousmonium: http://www.inagrm.com/accueil/concerts/lacousmonium

[2] R. Chandrasekhar, J. Goßmann, and L. Brümmer. "The ZKM klangdom." *Proceedings of the 2006 conference on New interfaces for musical expression*. Paris, France, 2006.

[3] the Huddersfield Immersive Sound System: http://www.thehiss.org

[4] J. Harrison, "Sound, space, sculpture: some thoughts on the 'what','how'and 'why'of sound diffusion." *Organised Sound* 3.02 (1998): 117-127.

[5] O. Chapman and K. Sawchuk. "Research-Creation: Intervention, Analysis and "Family Resemblances". *Canadian Journal of Communication* Vol 37 (2012) 5-26.

[6] An Interview with David Tudor by Teddy Hultber: http://davidtudor.org/Articles/hultberg.htm

[7] H. Vaggione, "L'espace Composable. Sur Quelques Catégories Opératoires Dans La Musique Électroacoustique", *L'espace, Musique /philosophie,* Archives Karéline. Paris, 1998. p.154.

[8] A. Di Scipio, "'Sound is the interface': from interactive to ecosystemic signal processing." *Organised Sound* 8.03 (2003): 269-277.

[9] Pierre Bastien: http://www.pierrebastien.com/en/biography.php

[10] "Stimuline":http://www.bipolar-production.com/stimuline-pook-clauss/?lang=en

[11] S. Merchel, M.E. Altinsoy and M. Stamm, "Touch the Sound: Audio-Driven Tactile Feedback for Audio Mixing Applications", *Journal of the Audio Engineering Society*, 60(1/2), pp. 47-53, 2012.

[12] O. Lähdeoja, A. Haapaniemi and V. Välimäki, "Sonic scenography - equalized structure-borne sound for aurally active set design", *Proceedings of the ICMC|SMC|2014*, Athens, Greece, 2014.

Hand Gestures in Music Production

Axel Berndt, Simon Waloschek, Aristotelis Hadjakos
Center of Music and Film Informatics
University of Music Detmold, Detmold, Germany
{berndt|waloschek|hadjakos}@hfm-detmold.de

ABSTRACT

Today's music production interfaces are still dominated by traditional concepts. Music studios are a prime example of a multi-device environment with some of the most complex user interfaces. In this paper, we aim at introducing a more embodied interaction modality, freehand interaction, to this scenario. Not as replacement! We analyze typical music production scenarios and derive where freehand input yields added value. We give an overview of related research and discuss aspects of the technical integration into music studio setups. This is complemented by a prototype implementation and a survey that provides clues for the development of an intuitive gesture set for expressive music control.

1. MOTIVATION

Music studios were always multi-device environments with specialized equipment for literally every task, such as sound synthesis, audio effects, recording, and mixing. At the center of this network lies the *Digital Audio Workstation* (DAW), a computer that integrates all hardware components and software supplements. Despite this high degree of specialization and distribution of functionality over a multitude of networked components we still observe a predominant uniformity of user interface concepts which are rooted in classic hardware devices. While new interface technologies are vitally incorporated in the context of digital musical instruments, we cannot register a similar fertilization in the field of music production. One main reason for this is a considerable interference with established optimized workflows. While each developer in this field is confronted with this problem, their thoughts and solutions were neither sufficiently documented nor discussed so far.

This paper addresses freehand gesture input. Our aim is not an entire replacement of established input modalities. Knobs, sliders, keyboard and mouse proved their worth and can be regarded as optimal solutions for many tasks. We want to keep established workflows intact. The greatest gain of freehand input lies in the continuous control of big ranges of parameters that develop over time, live with the musical playback. The control data that derives from the hand gestures can be converted to MIDI data. If music is produced solely electronically, without recorded human musicians, this may be used to steer expressive parameters such as tempo, dynamics, articulation, and timbre. This is our target scenario in this text. First, we give an overview of exemplary previous and related work in this field. The technical integration into the typical DAW setup is discussed in section 3. In section 4 we report of a survey to gain first cues for developing an intuitive set of gestures.

2. HANDS ON MUSIC: RELATED WORK

Many approaches to freehand gesture-controlled expression are based on the tracking of hand-held devices such as batons [1], drumsticks [2], and balls [3]. These may not only be tracked externally but may be equipped with sensors themselves, such as the Wii Remote, smartphones [4] and others [5, 6]. To avoid the necessity of holding a device, sensors can be fixed at the hand [7] or data gloves can be used [8, 9]. With optical tracking systems such as the HandSonor system [10], Microsoft's Kinect, and the Leap Motion no hand-held devices are necessary at all.

Typical software frameworks for musical freehand interaction are the commercial GECO system [11] and the Harmonic Motion toolkit [12]. The authors of the latter performed a preparatory survey of potential users. One outcome was the fact that 57% of their participants saw the most problematic issue of gestural music interaction in the mapping of gesture data to musical data. Speed, latency, hardware-specific problems, and stability came far after.

Wanderley [13] lists five different musical contexts of gestural control in music of which the following three are directly related to the process of music making and production and will be in the focus of the subsequent overview of related work in the field. *Instrument manipulation* takes place at the level of realtime sound synthesis control of digital musical instruments. *Score-level control*, such as conducting, manipulates the expression of a performance, not the material performed. *Post-production activities* address the control of digital audio effects, sound spatialization, and mixing. The large corpus of works in this field cannot be recapitulated completely here. We will pinpoint representative and illustrative works. A more comprehensive treatment of the subject can be found in [14, 15].

Sridhar [10] distinguishes continuous and discrete gesture data which can be mapped to likewise continuous and discrete instrument parameters. Dahl [16] focusses on the latter, when he studies air drumming gestures to identify motion features that provide the best timing information for discrete musical triggering. Françoise et al. [17] bypass the necessity of an explicit definition of the gesture-to-sound mapping by a machine learning method.

Gossmann & Neupert [18] introduce the *remix instrument*. Recorded audio material of an instrument is analyzed and its atomic sounds are arranged in a 3d scatter plot. The user's hand moves the playback cursor of a concatenative synthesis through this 3d space. The artistic installation, "Non Human Device #002" [19], allows for the control of sound parameters through freehand interaction with a jellyfish-like virtual creature. Two further instruments, the Air-Keys and the Air-Pads [20], are critically discussed in a lessons-learned report giving practical hints on playability and tracking performance for such projects. Further mappings for effects and synthesis modulation are described by Hantrakul & Kaczmarek [21].

The VUZIK/ChoirMob interface [22] performs a predefined polyphonic score. The performers manipulate the expression of the synthesized vocal sounds via touch gestures on smartphones. This work demonstrates the coupling of predefined musical "raw material" and its expressive realtime manipulation through gestures. Such predefined musical material does not necessarily have to be static but can also be generated in realtime. Tormoen et al. [23] use the Leap Motion controller in combination with the Rubato Composer for gesture-based music composition and improvisation, with particular regard to the transformation of musical material. Such interactive, semi-automatic composing and performance systems constitute a seamless fade between instrument- and score-level interaction.

Hand tracking-based music conducting systems represent a popular type of score-level applications. Lim & Yeo [4] track conducting gestures via the gyroscope sensor of a smartphone. The MICON system [24] is an interactive exhibit that optically tracks a baton with an infrared sensor.

Gesture-controlled audio mixing is another recurrent subject [7, 25, 26]. Balin & Loviscach [27] see the chance of reducing the complexity of traditional DAW's GUIs via gestural control elements. They developed and evaluated a mouse and touch-operated gesture set. Ratcliffe [28] visualizes audio tracks as spheres in a 3d space that can be grasped for stereo positioning and depth mixing. A similar so-called *stage metaphor* has been adopted by Lech & Kostek [29] who further propose a comprehensive set of symbolic hand gestures. This may further help to alleviate the attention dragging, adulterating visuality of traditional DAW GUIs [29, 30].

All these works show freehand gesture control being an interaction modality that holds several promising perspectives for music production beyond the pure instrument playing. This requires both, its introduction to established workflows and the development of appropriate gesture sets, including corresponding mappings. Not all functionality in music production benefits from this input type. For some functions faders, knobs, mouse, and keyboard are an optimal choice. Freehand gestures should rather be seen as a complement than a replacement. The following section puts the focus on the technical side and discusses the integration of freehand gesture control in typical DAW setups.

3. HANDS ON THE DAW

This section discusses several noteworthy aspects of the technical integration of freehand interaction into a typical DAW environment. Figure 1 gives an overview of the resulting architecture.

Several hand tracking solutions are available, today. Very popular is Microsoft's Kinect 2. Its comparatively low resolution and frame rate of about 30fps, however, make it more suitable for expansive whole-body gestures and a rather rough control of musical parameters, e.g. slow and steady dynamics and tempo changes. Fast fine-grained control, such as note-wise articulations, are impractical. The Leap Motion controller, on the other hand, is specialized for hand tracking. Its tracking range is distinctly smaller compared to the Kinect but offers a superior resolution that allows for very fine-grained hand poses and gestures. Even very fast gestures can be detected reliably thanks to its sampling rate of up to 300fps. Since it was designed for the use on tables, it seems the device of choice for professional audio workstations and is small enough to easily find a place in-between the other devices which is a key advantage over many other tracking systems that involve several cameras distributed around the tracking area. The user can quickly switch between gestural input and any other device. Especially when using the non-dominant hand for gesture input while keeping the dominant hand on a primary device seems advantageous in this scenario.

Next, a gesture recognizer identifies meaningful gestures. A mapping process converts these into a format that modern DAWs can further process and record, e.g. MIDI. Inside the DAW, the MIDI data can be used to control various parameters of sound synthesis and effect plugins as well as the overall mix. Frameworks such as the commercial GECO system [11] allow users to define their own mappings. For controlling solely sound-related parameters of a given musical material, the software chain ends here. However, this does not allow for more complex control of tempo, micro timing or alteration of the note material. More sophisticated tasks, e.g. gesture-driven realtime algorithmic music generation, require an additional, more flexible MIDI sequencer. In such a case, the DAW is only used as a sound and effects generator and recording device. The standalone sequencer takes over the responsibility to read, modify and even generate MIDI data.

Direct feedback during input generally eases the process of learning new input modalities and reduces users' mental load. The user should get notified about not or wrongly detected gestures instead of being frustrated by opaque decisions of the gesture recognizing system. Therefore, the visualization of tracking data (body, hands, depth sensor, gestures) as well as the audio output and additional auditory cues, presented in realtime, are advisable and allow for quick reactions and adaptations by the user. Such feedback requires a low latency to the gestural input. This requirement may be relaxed in non-live situations where no discrete and precisely timed sound events have to be entered.

4. SURVEY ON GESTURES

As we have pointed out previously freehand interaction in music production scenarios has, in our opinion, its greatest potential in the control of musical parameters that are otherwise laborious to handle, in particular multiple continuous parameters at once and live with the musical playback. Typical candidates for this are sound synthesis and audio effects parameters as well as expressive performance

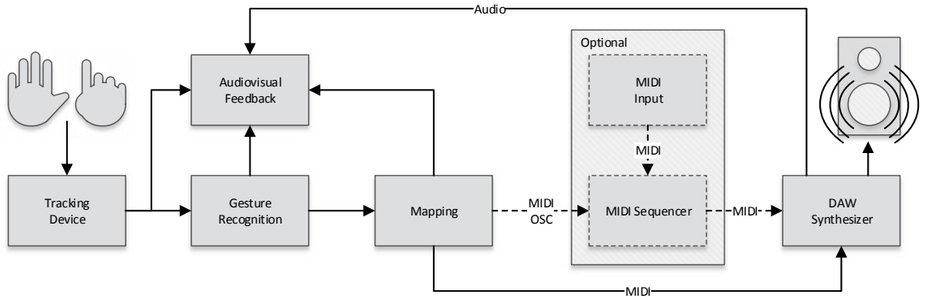


Figure 1. Integration of freehand interaction into a DAW.

parameters (assuming that the music is partly or solely produced on the computer). The range of such, possibly interesting parameters is wide. The concept of digital musical instruments is related to the application scenarios addressed but additionally involve the triggering of musical events. In contrast to this, we regard the musical raw material, i.e. the notes, as fixed, but not the way they are performed. Multi-channel mixing and conducting have already been addressed by many others. Hence, we decided to focus on the expressive performance, i.e. timing, dynamics, articulation, and timbre, which were considered only rarely in previous work so far.

In search of an intuitive gesture set we conducted a survey. The participants were asked to suggest gestures they find intuitive to manipulate tempo (faster/slower), dynamics (louder/softer), articulation (legato/staccato), and timbre (brighter/darker or shriller/softer). This covers only a subset of the manifold possibilities to manipulate expressive music performances which cannot all be included in only one single survey. Hence, we decided to focus on the most prominent features first and see if the supposed gestures may already be applicable also to more fine-grained features such as metrical accentuation and rubato. A follow-up survey can then focus on these and invite participants with a respectively more professional background.

4.1 Setup, Participants, Conduct

The survey took place during an open house event at Ostwestfalen-Lippe University of Applied Sciences at May 9th 2015 in Lemgo, Germany from 10am to 4pm. The location was a seminar room (doors kept open) equipped with speakers and projector. These were used to operate a prototype implementation. Here the visitors got used to the Leap Motion controller and could produce sound output with hand gestures. Besides the more playful wind and laser sword-like sounds the users could manipulate live-generated music (homophonic chorale) according to the four parameters of the survey. The predefined one-hand gestures—actually poses—were as follows:

Tempo was controlled with the depth position of the hand. Moving the hand forward increases the musical tempo, putting the hand back slows the tempo down.

Dynamics derived from the vertical hand position. Moving the hand up and down causes the volume level to increase and decrease.

Articulation was controlled by the grab strength. The flat hand produced a legato and the fist a staccato articulation. For poses in-between the note lengths were interpolated.

Timbre manipulation was done by hand tilting between the horizontal (soft sound) and vertical (shrill sound) pose. The timbres were achieved by oscillator hard syncing.

All parameters are controlled at once. The horizontal axis was left unused to be able to take the hand out of the tracking frustum and keep a certain parameter setting.

Among the numerous visitors, 44 took part in the interview (23 male, 21 female) aged from preschool to retirement age. None of them had a professional music background, hence there was few bias towards classic conducting gestures. Some participants got to know the prototype demo already before the interview. In these cases their answers could have been biased by the predefined gestures. In case that they answered these same gestures we insisted in further suggestions and counted the demo gesture suggestion in the evaluation only if the participants still explicitly preferred it over the own suggestions. The interviews were video recorded from the front. In the evaluation we collected all gestures and counted the suggestions.

4.2 Results & Discussion

We collected 281 suggestions of 115 gestures including those that are repeatedly suggested for different tasks. In some cases, the participants suggested gestures that indicate a certain parameter value, e.g. tempo specification by beat knocking. As this implicitly includes increase and decrease, e.g. knocking faster or slower, we counted these suggestions twice, i.e. once for increase and once for decrease. We identified 47 gesture pairs, i.e., two identical but inversely performed gestures.

Tempo Gestures: For tempo control we had 71 suggestions of 31 gestures. Only 14 suggestions (19.7%) of 10 gestures involved both hands. 26 suggestions (36.6%) of 9 gestures were actually poses that specify the tempo, e.g., through grab strength or the vertical position of the hand. All others indicate the tempo through their motion characteristics. The demo gesture (quasi a forward/ backward lever) was never suggested. The top-rated gestures are shown in table 1.

Further variants of the "calm-down gesture" $t1$ differed by using one or two hands, orienting the palm generally

Accelerando (increase tempo, 100% = 29 suggestions)

$T1$	fast fanning away with the back of one (open) hand, also described as wiping away (9 suggestions, 31%)
$T2$	fast circular movement of one (open) hand, away from the body, back of the hand heading forward (5 suggestions, 17.2%)
$T3$	one hand moves to the right, also described as fast-forwarding on a video recorder (4 suggestions, 13.8%)

Ritardando (decrease tempo, 100% = 34 suggestions)

$t1$	one (open) hand, palm downward, moves downward, also described as "calm-down gesture" (11 suggestions, 32.4%)
$t2$	opposite of gesture $T3$, one hand moves to the left, also described as rewinding on a video recorder (4 suggestions, 11.8%)

Table 1. Suggested gestures for musical tempo control.

downward or into the direction of movement and use of a decent shaking to activate or intensify the input. In all variants the downward movement of the hands was used for slowdown and upward movement for acceleration. Even though each variant was suggested only once or twice, altogether (including the above "calm-down gesture") we had 18 suggestions (25.4% of all tempo suggestions).

We had also 18 suggestions (25.4% of all tempo suggestions) of "beat knocking" gestures (up and down movement) in different variants, including one or two symmetrical moving hands, open hand or stretched index finger, hand clapping, and other rhythmic hand motion. Here, the tempo derives from the pace of the motion pattern.

Dynamics Gestures: For dynamics control the participants had 93 suggestions of 25 gestures, including 81 suggestions (87.1%) of 19 poses and 36 suggestions (38.7%) of 9 bimanual gestures. This reflects a preference of one-handed poses for dynamics control. Table 2 shows the top-rated dynamics gestures.

Most of these gestures are variants of gesture $D1$ and $d1$, respectively. In sum, we got 65 suggestions (70% of all dynamics suggestions) for vertical hand movement, upward to increase and downward to decrease the volume level. These were already implemented in the demo and, as far as the participants knew them, widely confirmed.

Articulation Gestures: The participants gave 66 suggestions of altogether 21 gestures. This includes 23 suggestions (34.8%) of 10 poses and 25 suggestions (37.9%) of 7 bimanual gestures (top-rated gestures in table 3).

Gestures $a1$ and $a2$ were always jointly suggested. Their only difference is the use of one or both hands. Thus, we see them as equivalent. We further observed that the participants preferred gestures that involve grabbing/finger spreading and smooth/choppy movements to indicate articulation (53 of 66 suggestions, 80.3%).

Timbre Gestures: This musical feature was perceived as the most difficult to express through gestures. This mirrors not only in many oral comments but also in a greater diversity of suggested gestures (all together 51 suggestions of 38 gestures) and a low maximum score of 4. We collected 36 suggestions (70.6%) of 27 poses and 15 sugges-

Crescendo (increase volume level, 100% = 45 suggestions)

$D1$	one (open) hand, palm downward, moves upward (17 suggestions, 37.8%)
$D2$	both hands open, palms facing each other, spreading arms. In some cases the movement triggers the crescendo irrespective of the distance of both hands. Other participants expressed a specific loudness value through the distance between both hands. (5 suggestions, 11.1%)
$D3$	similar to $D1$ but with palm heading upward (4 suggestions, 8.9%)
$D4$	similar to $D1$ but with two hands (4 suggestions, 8.9%)
$D5$	similar to $D1$ but with two hands and palm heading upward (4 suggestions, 8.9%)
$D6$	one (open) hand, held vertical, moves to the right (3 suggestions, 6.7%)

Decrescendo (decrease volume level, 100% = 48 suggestions)

$d1$	opposite of gesture $D1$, one (open) hand, palm downward, moves downward (18 suggestions, 37.5%)
$d2$	opposite of gesture $D4$, similar to $d1$ but with both hands (16 suggestions, 33.3%)
$d3$	opposite of gesture $D2$, both hands open, palms facing each other, bringing arms together (3 suggestions, 6.3%)
$d4$	opposite of gesture $D6$, one (open) hand, held vertical, moves to the left (3 suggestions, 6.3%)

Table 2. Suggested gestures for musical dynamics control.

tions (29.4%) of 12 bimanual gestures. This indicates one-handed poses to be the preferred gesture types to control timbre. The top-rated gestures are listed in table 4.

22 suggestions (34.1%) specifically involved the fingers in some way, be it in the form of fast, chaotic or slow, wavy finger movement or the fingers' position (claw-like, flat, spread, or in right angle with the palm). Such a variety was not observed for the other musical parameters.

Discussion: Although we asked those participants who suggested gestures from the demo to make further suggestions and think about what they find intuitive, bias cannot entirely be excluded. On the other side, some of the demo gestures were not even mentioned or only once (specifically the gestures for tempo and timbre control). This fact suggests that the bias was not strong and/or the corresponding demo gestures not very successful.

We generally observed a preference of one-handed gestures. Only 90 suggestions (13.5%) out of 281 involved both hands. Regarding the typical music production workstation, where the user sits at a computer or mixer, the one-handed input is advantageous. Here, the user can keep one hand at the primary device and make freehand input with the secondary hand "by the way". This is also a good starting point for introducing multimodal interaction concepts.

Our results also include concurrent gestures, i.e. similar gestures for different tasks (e.g., $t1 = d1 = s3$, $T3 = D6$, $t2 = d4$, and $S3 = D1$). Hence, gesture combinations for parallel control of all four parameters are not possible with only the top-rated gestures. Instead, we will have to find a

Legato (broad articulation, 100% = 24 suggestions)

$A1$	smooth horizontal wave movement of one open hand, palm heading downward (6 suggestions, 25%)
$A2$	both hands open and arms wide open, also described as indicating a long tone length (5 suggestions, 20.8%)
$A3$	one hand, open/flat (4 suggestions, 16.7%)

Staccato (short articulation, 100% = 42 suggestions)

$a1$	rhythmic dabbing movement with fist, beak pose or thumb and index finger with one hand (11 suggestions, 26.2%)
$a2$	similar to $a1$ but with both hands moving symmetrical (11 suggestions, 26.2%)
$a3$	opposite of $A2$, both hands open, held close to each other, also described as indicating a short tone (4 suggestions, 9.5%)
$a4$	opposite of $A3$, fist (4 suggestions, 9.5%)
$a5$	opposite of $A1$, one (open) hand, vertically held, makes choppy up and down movements, also described as hacking with the side of the hand (4 suggestions, 9.5%)

Table 3. Suggested gestures for musical articulation control.

Bright/shrill (100% = 24 suggestions)

$S1$	spread fingers of one hand (4 suggestions, 16.7%)
$S2$	fast chaotic finger movements of one hand (2 suggestions, 8.3%)
$S3$	one open hand, palm heading downward, moves upward (2 suggestions, 8.3%)
$S4$	two claw hands, palms heading downward (2 suggestions, 8.3%)

Dark/soft (100% = 27 suggestions)

$s1$	smooth horizontal wave movement of one open hand, palm heading downward (3 suggestions, 11.1%)
$s2$	opposite of $S1$, one flat hand with closed fingers (2 suggestions, 7.4%)
$s3$	opposite of $S3$, one open hand, palm heading downward, moves downward (2 suggestions, 7.4%)
$s4$	both hands open, palms heading towards the computer, shaking, also described as repellent gesture to soften a shrill sound (2 suggestions, 7.4%)
$s5$	cover ears with both hands to soften the shrill sound (2 suggestions, 7.4%)
$s6$	swivel both open hands at the ears (2 suggestions, 7.4%)

Table 4. Suggested gestures for timbre control.

good trade-off in our further steps. An approach might be that we implement the gestures not exactly as suggested but adopt certain of their characteristics (use of vertical hand position, work with fingers, pose or motion etc.) and define a new set of combinable gestures on this basis.

5. SUMMARY

Music production takes place in multi-device environments. Highly specialized hard- and software modules mold an often complex architecture. We discussed the role and integration of freehand gesture input in this scenario. Beyond the traditional interfaces that proved well for many tasks we regard freehand input a beneficial complement whenever it comes to continuous realtime control of multiple expressive parameters, e.g., for sound synthesis, audio effects and expressive performance.

As a first step toward the development of an appropriate set of gestures we conducted a survey with 44 participants. Besides the clear preference of one-handed gestures we collected several clues on which aspect of hand gestures (vertical hand movement, grab gesture and other finger movements, palm rotation) are favored for which type of musical parameter.

Acknowledgments We like to thank all visitors of the open house event who participated in the interview. This project is funded by the German Federal Ministry of Education and Research (BMBF 01UG1414A-C).

6. REFERENCES

[1] M. V. Mathews, "Three dimensional baton and gesture sensor," United States Patent Nr. 4,980,519, Dec. 1990.

[2] C. Havel and M. Desainte-Catherine, "Modeling an Air Percussion for Composition and Performance," in *NIME 2004*. Hamamatsu, Japan: Shizuoka University of Art and Culture, June 2001, pp. 31–34.

[3] L. Stockmann, A. Berndt, and N. Röber, "A Musical Instrument based on Interactive Sonification Techniques," in *Audio Mostly 2008*. Piteå, Sweden: Interactive Institute/Sonic Studio Piteå, Oct. 2008, pp. 72–79.

[4] Y. K. Lim and W. S. Yeo, "Smartphone-based Music Conducting," in *NIME 2014*. London, UK: Goldsmiths, University of London, June 2014, pp. 573–576.

[5] C. Kiefer, N. Collins, and G. Fitzpatrick, "Evaluating the Wiimote as a Musical Controller," in *ICMC 2008*. Belfast, Northern Ireland: ICMA, Sonic Arts Research Centre, Queen's University Belfast, 2008.

[6] N. Rasamimanana, F. Bevilacqua, N. Schnell, F. Guedy, E. Flety, C. Maestracci, B. Zamborlin, J.-L. Frechin, and U. Petrevski, "Modular Musical Objects Towards Embodied Control of Digital Music," in *TEI 2011*. Funchal, Portugal: ACM, 2011, pp. 9–12.

[7] K. Drossos, A. Floros, and K. Koukoudis, "Gestural User Interface for Audio Multitrack Real-time Stereo Mixing," in *Audio Mostly 2013*, Interactive Institute/Sonic Studio Piteå. Piteå, Sweden: ACM, Sept. 2013.

[8] T. Mitchell, S. Madgwick, and I. Heap, "Musical Interaction with Hand Posture and Orientation: A Toolkit of Gestural Control Mechanisms," in *NIME 2012*. Ann Arbor, Michigan, USA: University of Michigan, School of Music, Theatre & Dance, May 2012.

[9] S. Serafin, A. Trento, F. Grani, H. Perner-Wilson, S. Madgwick, and T. Mitchell, "Controlling Physically Based Virtual Musical Instruments Using The Gloves," in *NIME 2014*. London, UK: Goldsmiths, University of London, June 2014, pp. 521–524.

[10] S. Sridhar, "HandSonor: A Customizable Vision-based Control Interface for Musical Expression," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. Paris, France: ACM, Apr./May 2013, pp. 2755–2760.

[11] Uwyn bvba/sprl, "GECO: Multi-Dimensional MIDI/OSC/CopperLan Expression Through Hand Gestures," app on Airspace store, 2013, version 1.3.0.

[12] T. Murray-Browne and M. D. Plumbley, "Harmonic Motion: A Toolkit For Processing Gestural Data For Interactive Sound," in *NIME 2014*. London, UK: Goldsmiths, University of London, June 2014, pp. 213–216.

[13] M. M. Wanderley, "Gestural Control of Music," in *Proc. of the Int. Workshop on Human Supervision and Control in Engineering and Music*, Kassel, Germany, 2001, pp. 101–130.

[14] R. I. Godøy and M. Leman, Eds., *Musical Gestures: Sound, Movement, and Meaning*. New York, NY, USA: Routledge, Feb. 2010.

[15] E. R. Miranda and M. M. Wanderley, *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*, ser. The Computer Music and Digital Audio Series. Middleton, WI, USA: A-R Editions, Inc., July 2006, vol. 21.

[16] L. Dahl, "Triggering Sounds From Discrete Air Gestures: What Movement Feature Has the Best Timing?" in *NIME 2014*. London, UK: Goldsmiths, University of London, June 2014, pp. 201–206.

[17] J. Françoise, N. Schnell, and F. Bevilacqua, "A Multimodal Probabilistic Model for Gesture-based Control of Sound Synthesis," in *Proc. of the 21st ACM Int. Conf. on Multimedia*. Barcelona, Spain: ACM, 2013, pp. 705–708.

[18] J. Gossmann and M. Neupert, "Musical Interface to Audiovisual Corpora of Arbitrary Instruments," in *NIME 2014*. London, UK: Goldsmiths, University of London, June 2014, pp. 151–154.

[19] R. Carvalho and M. Neto, "Non Human Device #002," in *xCoAx 2014: Proc. of the 2nd Conf. on Computation, Communication, Aesthetics and X*, M. Carvalhais and M. Verdicchio, Eds., Faculdade de Belas Artes. Porto, Portugal: Universidade do Porto, June 2014, pp. 490–492.

[20] J. Han and N. Gold, "Lessons Learned in Exploring the Leap Motion™ Sensor for Gesture-based Instrument Design," in *NIME 2014*. London, UK: Goldsmiths, University of London, June 2014, pp. 371–374.

[21] L. Hantrakul and K. Kaczmarek, "Implementations of the Leap Motion in sound synthesis, effects modulation and assistive performance tools," in *Joint Conf. ICMC and SMC*, University of Athens. Athens, Greece: International Computer Music Association, Sept. 2014, pp. 648–653.

[22] A. Pon, J. Ichino, E. Sharlin, D. Eagle, N. d'Alessandro, and S. Carpendale, "VUZIK: A Painting Graphic Score Interface for Composing and Control of Sound Generation," in *ICMC 2012*, M. M. Marolt, M. Kaltenbrunner, and M. Ciglar, Eds. Ljubljana, Slovenia: International Computer Music Association, University of Ljubljana, Sept. 2012, pp. 579–583.

[23] D. Tormoen, F. Thalmann, and G. Mazzola, "The Composing Hand: Musical Creation with Leap Motion and the BigBang Rubette," in *NIME 2014*. London, UK: Goldsmiths, University of London, June 2014, pp. 207–212.

[24] J. Borchers, A. Hadjakos, and M. Mühlhäuser, "MICON a Music Stand for Interactive Conducting," in *NIME 2006*. Paris, France: IRCAM – Centre Pompidou, June 2006, pp. 254–259.

[25] M. T. Marshall, J. Malloch, and M. M. Wanderley, "Gesture Control of Sound Spatialization for Live Musical Performance," in *Gesture-Based Human-Computer Interaction and Simulation*, ser. Lecture Notes in Computer Science, M. Sales Dias, S. Gibet, M. M. Wanderley, and R. Bastos, Eds. Berlin, Heidelberg, Germany: Springer Verlag, 2009, vol. 5085, pp. 227–238.

[26] P. Quinn, C. Dodds, and D. Knox, "Use of Novel Controllers in Surround Sound Production," in *Audio Mostly 2009*. Glasgow, Scotland: Glasgow Caledonian University, Interactive Institute/Sonic Studio Piteå, Sept. 2009, pp. 45–47.

[27] W. Balin and J. Loviscach, "Gestures to Operate DAW Software," in *130th Audio Engineering Society Convention*. London, UK: Audio Engineering Society, May 2011.

[28] J. Ratcliffe, "Hand Motion-Controlled Audio Mixing Interface," in *NIME 2014*. London, UK: Goldsmiths, University of London, June 2014, pp. 136–139.

[29] M. Lech and B. Kostek, "Testing A Novel Gesture-Based Mixing Interface," *Journal of the Audio Engineering Society*, vol. 61, no. 5, pp. 301–313, May 2013.

[30] B. Owsinski, *The Mixing Engineer's Handbook*, 3rd ed. Boston, MA, USA: Course Technology, Cengage Learning, 2014.

GRAB-AND-PLAY MAPPING: CREATIVE MACHINE LEARNING APPROACHES FOR MUSICAL INCLUSION AND EXPLORATION

Hugo Scurto
Department of Computing
Goldsmiths, University of London
h.scurto@gold.ac.uk

Rebecca Fiebrink
Department of Computing
Goldsmiths, University of London
r.fiebrink@gold.ac.uk

ABSTRACT

We present the first implementation of a new tool for prototyping digital musical instruments, which allows a user to literally grab a controller and turn it into a new, playable musical instrument almost instantaneously. The tool briefly observes a user interacting with a controller or sensors (without making any sound), and then it automatically generates a mapping from this observed input space to the control of an arbitrary sound synthesis program. The sound is then immediately manipulable using the controller, and this newly-created instrument thus invites the user to begin an embodied exploration of the newly-created relationships between human movement and sound. We hypothesize that this approach offers a useful alternative to both the creation of mappings by programming and to existing supervised learning approaches that create mappings from labeled training data. We have explored the potential value and trade-offs of this approach in two preliminary studies. In a workshop with disadvantaged young people who are unlikely to learn instrumental music, we observed advantages to the rapid adaptation afforded by this tool. In three interviews with computer musicians, we learned about how this "grab-and-play" interaction paradigm might fit into professional compositional practices.

1. INTRODUCTION

Historically, computer programming has been a core technique used in the creation of new digital musical instruments. The "mapping" [1] that specifies how a musician's movements (sensed using a controller or sensors) relate to sound (e.g., the values of sound synthesis parameters) is often created by writing programming code. While programming allows a mapping to be specified precisely, the process of translating an intended mapping function to code can be frustrating and time consuming [2], even for expert programmers, and it is inaccessible to non-programmers.

Machine learning has been used as an alternative mechanism for generating mappings since the early work of [3]. Most work that has employed machine learning for mapping creation has employed supervised learning algorithms, which can create a mapping from input sensor values to sound synthesis control parameters using a set of "labeled" training examples. In this labeled training set, each example consists of one vector of input sensor values, plus the "labels"—the vector of sound synthesis parameter values the designer would like to be produced in response to those sensor values. Research has suggested that supervised learning offers a useful alternative to programming, for instance by making mapping creation faster, by enabling designers to encode an embodied understanding of the desired gesture/sound relationships in the training examples, and by making mapping accessible to non-programmers [2, 4].

However, existing supervised learning approaches to mapping creation do not directly address some of the most fundamental needs of instrument designers. For instance, an instrument designer often does not know a priori precisely what type of mapping she wants in a new instrument. It is only by prototyping—experimenting with alternative designs in a hands-on way—that she can more fully understand the potential offered by a set of sensors and synthesis tools, and understand how she might fit these together into an instrument or a performance. An instrument designer who wants to explore many different prototypes using machine learning must still create many different sets of training data, and explicitly choose the type of relationship between sensors and sounds that should be embedded within each set.

New approaches to mapping generation might accelerate the discovery and realisation of new design ideas, by taking advantage of the computer's ability to generate mapping functions under different types of constraints or with different types of goals. This could be useful when the user does not have a specific relationship between movement and sound already in mind, or when other properties of the instrument (e.g., playability, comfort) supersede any preference for a particular sensor/sound relationship.

In this paper, we describe first steps toward exploration of such alternative mapping strategies. We have implemented a fully-functioning tool capable of generating many alternative mappings from a single set of unlabeled training examples, which encode the range of motion of a performer using arbitrary sensors/controllers. In our first version of the system, alternative mappings are generated from this single training set using a computationally straightforward approach to transform the unlabeled training set into multiple alternative labeled training sets, which can each be used to build a mapping using supervised learning. Many other computational approaches to generating multiple alternative mappings from unsupervised learning are also possible.

We have worked with two sets of users to evaluate this approach and better understand its potential use. These users include youth with disabilities and difficult life circumstances, as well as three professional computer music composers. This work suggests that the rapid adaptation afforded by this approach could benefit the first category of users, while the predisposition to musical exploration and discoveries could benefit the second category.

2. BACKGROUND AND PREVIOUS WORK

Machine learning algorithms have been widely employed in musical interaction, both as a means to analyze musical gestures and to design gesturally-controlled digital musical instruments (see [5] for an overview of the field).

Research by Fiebrink and collaborators has focused on understanding the impact of using machine learning (as opposed to programming) on the instrument design process [2], and on designing user interfaces to allow instrument builders to use machine learning effectively and efficiently, without requiring prior machine learning expertise [4]. Fiebrink's Wekinator [1] toolkit allows instrument builders to create supervised learning training sets by demonstrating performer gestures alongside the instrument sounds the designer would like to be produced by those gestures. The Wekinator uses general-purpose algorithms for regression (e.g., multilayer perceptron neural networks, linear and polynomial regression) and classification (e.g., nearest-neighbor, support vector machines) to create mappings from this data.

Other recent research has explored the development of new modeling approaches that are tailored to building gestural musical interactions [6, 7], notably allowing for similarity estimations between a gesture being performed and recorded references. Such approaches are particularly successful when the task is to recognize and track given gestures.

There is a growing interest among music researchers in the importance of bodily experience in sound perception and cognition [8]. According to this theory, it is primarily through the body that performers convey inner information about their artistic intentions and emotions; this bodily information is encoded into and transmitted by sound to listeners who can in turn attune to the performer's inner intent. It is important to underscore that such body movements, or gestures, are not necessarily pre-defined for the performer, and can appear to be metaphorical [9] rather than descriptive [10, 11]. In this sense, mapping approaches that value exploration rather than explicit definition could be relevant to facilitate the use of metaphorical gestures in performance.

3. GRAB-AND-PLAY MAPPING

3.1 Definition

We propose a new paradigm for mapping creation, called "grab-and-play mapping", that enables the very rapid creation of new instruments from a very small amount of data communicating some minimal, soft design constraints—namely, the way the user might want to move while playing this new instrument. This minimal set of data allows the creation of mappings which are customised to a controller and/or to a player in a loose sense, by aiming for a mapping that is playable using whatever range of motion and types of variation are present in the examples provided by the designer. But this process does not require a designer to specify other information about the instrument, other than potentially the range of legal values for each sound synthesis parameter that will be controlled by the mapping. Our approach thus shifts the designer's focus from one of imagining and then implementing new gesture-sound relationships, to a focus on discovering new relationships that have been designed partly by the computer, and on embodied exploration of those relationships.

[1] www.wekinator.org

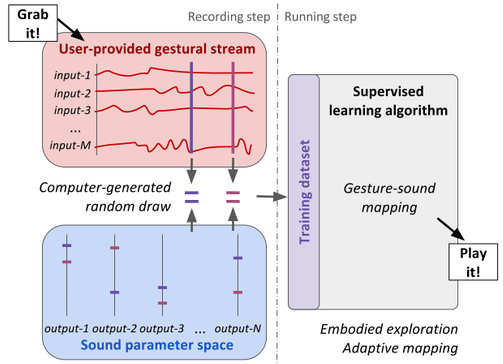


Figure 1. Our first implementation of the grab-and-play mapping paradigm. Inputs and outputs are respectively drawn from the user's recorded gestural stream and the sound parameter space. Outputs from 1 to N are sound synthesis parameters. In this schema, the training database contains two examples (i.e. two input-output pairs).

3.2 Implementation

Our vision of grab-and-play mapping could be implemented using a number of techniques for automatically generating a mapping. This paper reports on our first implementation, which is described in Figure 1.

In this implementation, the user must first demonstrate how she will physically interact with a musical controller; this results in a recorded, continuous stream of gestural input data. Next, the computer transforms this stream of unlabeled inputs into a labeled training set that can be passed to a supervised learning algorithm for mapping creation (e.g., a neural network). Specifically, a number of examples are chosen at random from the recorded inputs. Each example is assigned a randomly-generated value for each sound synthesis parameter. These random sound synthesis parameters could be chosen from user-selected "presets" (i.e., vectors of parameter values that, together, result in sounds the user might want to have present in the instruments). Or, each parameter could be randomly generated from a uniform distribution over the range of all legal parameter values (e.g., [0,1]).

Finally, this artificially-generated training set is fed into a supervised learning algorithm that builds a mapping function capable of computing a new sound synthesis parameter vector for any new control vector. The user can now play the newly-created instrument by interacting with the

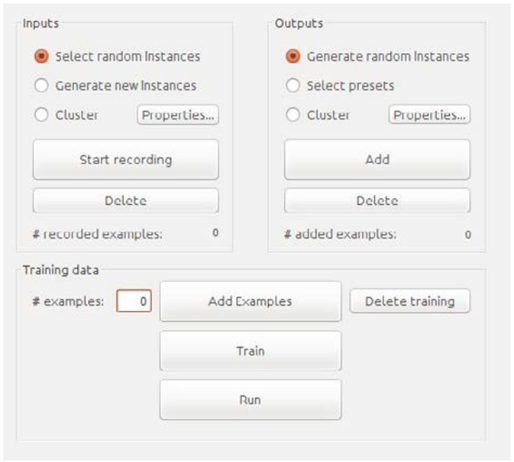


Figure 2. The current GUI of the tool. Observational studies reported in this paper only used the random implementation for input; random implementation and preset implementation for output. Other implementations of the grab-and-play approach are already implemented (see section 6), and will be studied in the near future.

input controller or sensors and discovering how the sound changes with her actions.

This new tool is implemented in Java as a branch of the Wekinator software. All code is available online [2]. The new tool adds the following additional functionality to Wekinator (see Figure 2):

- Grab-and-play mappings can be generated using the procedure above, requiring only that the user demonstrate a brief input sequence.
- New alternative mappings can be generated indefinitely from the same grab-and-play unlabeled training sequence.
- The user can interactively change the number of supervised training examples randomly generated from the grab-and-play training sequence.
- The user can switch between grab-and-play mapping and mappings generated using supervised learning.

The tool also takes advantage of the following existing capabilities of Wekinator:

- Any type of input controller or sensor system can be used to control sound, provided data about the input is sent as an OSC message [12].
- Any sound synthesis software can be used to play sound, provided it can receive synthesis parameter vectors as OSC messages.
- The GUI allows users to switch immediately and repeatedly between generating mappings and playing the generated instruments in real-time.
- The GUI allows users to easily change mappings by deleting and adding training examples.
- Advanced or curious users can customise aspects of the machine learning process, e.g., changing the learning algorithm or its parameters, changing the selected features, etc.
- Learning algorithms are set to default configurations that have been shown to work well for many mapping problems, so novice users never have to make an explicit choice of learning algorithm or algorithm parameters.

[2] http://github.com/hugoscurto/GrabAndPlayWeki

4. PRELIMINARY WORKSHOP WITH DISABLED YOUNG PEOPLE

We used this tool in a workshop with disabled young people to gain a preliminary understanding of how it might be useful for building new musical instruments for people with disabilities, and of how youth might respond to the customised yet unpredictable mappings built by this tool.

4.1 Using machine learning to build instruments with disabled people

Machine learning has been recently applied to build custom musical interfaces for disabled adults through several workshops [13]. Not only did the authors of that work find similarities between the musical goals and practices of disabled people and expert musicians, but they also noted some difficulties for participants to develop a memorable gestural vocabulary. In our workshops, we were therefore curious whether this grab-and-play approach might circumvent some user frustration, by explicitly inviting exploration of new instruments rather than suggesting that gesture design and memorisation are important.

4.2 Workshop setup

4.2.1 Participants

The workshop we led was the first workshop of a Musical Inclusion Programme [3], one of the aims of which is to help disadvantaged young people take part in musical activities. "Disadvantaged" stands for a broad variety of living conditions, ranging from health, behavior or learning disorders to looked-after children. Such young people may not have the opportunity to access high-quality musical activities, thus preventing them from the benefits music can provide in a social context. Bespoke digital musical instruments have the potential to make music-making easier and more fun for many of these youth. It is also possible that using personalised instruments may reduce social pressure, since the mapping function is unique to each user. By emphasising participation as a process of exploration of instruments and sound rather than performing a piece of music correctly, we also hoped to make the experience fun and inclusive for everyone.

The 15 youth we worked with all had physical and/or mental disabilities. They were accompanied by their parents or guardians, and their level of concentration was variable depending on their disabilities.

4.2.2 Workshop structure

The workshop was a one-hour session during which each of the two workshop leaders led a sequence of small-group sessions with one youth participant and their parent/guardian(s).

The input device used was a GameTrak Real World Golf controller, which senses 3D position of the user's hands using two strings. Sound was generated by Max/MSP. The following setups were available to the participants:

- Grab-and-play classification for triggering pre-recorded sound samples, in a "funk" style.
- Grab-and-play regression for controlling audio effects (pitch shifting and reverb).
- The same sample triggering and effects control as above, but using Wekinator's existing supervised learning interfaces for classification and regression (i.e., requiring users to specify labeled training examples).

In each small group, the workshop leader controlled the computer (including the GUI for mapping creation), and the youth participant was given the input controller (sometimes with the help of parent/guardians). Participants therefore did not have to learn to use the GUI or other software. All participants tried at least one grab-and-play mapping, and participants who had time and expressed interest also tried supervised learning mapping.

[3] http://www.nmpat.co.uk/music-education-hub/Pages/musical-inclusion-programme.aspx

4.3 Observational study

4.3.1 Grab-and-play setup

Our grab-and-play approach was very useful to build adapted interfaces. It allowed us to build instruments whose gestural range was wholly dependent on the participant: during the recording step, some people made wide movements, while others with strong motor disabilities were only able to make small movements. In this sense, the adaptivity of our tool prevented it from building non-playable instrument for a given person. Some participants also seemed to find the exploratory side of the running step very fun. They spent a lot of time trying to find where the different audio samples were in their gestural space: this activity seemed to capture participants' attention, as they usually seemed to engage in choosing which sample to trigger.

Grab-and-play classification seemed to elicit different types of interaction compared to regression. People using classification focused on triggering known sounds, whereas people using regression focused on exploration (alternating between searching for new sounds and playing them back). Both approaches thus have their own pros and cons, depending on which musical activity people and carers want to take part in.

4.3.2 Original Wekinator setup

Participants who had enough concentration also tried the supervised learning setup. They first recorded different GameTrak positions for each of the four classes of samples, and then tried their instrument. Several participants reported that they liked being able to choose where to place the audio samples in their gestural space, giving them even more control on what was going on. However, it was hard for some participants to concentrate on the process of choosing different gestures to trigger different samples. Even if the customization of the interface was enjoyed by some participants, it was not necessary to support meaningful musical experiences for most participants.

Both classification and regression were understood by participants, as they knew which audio effect to expect since they had chosen them during the recording step.

4.3.3 General discussion

This preliminary workshop has shown the utility of our grab-and-play approach to build custom musical interfaces. Our observations show this approach can be useful to build personalised devices, both for participants that were not able to concentrate for a long time, and for participants with specific motor disabilities. In any case, using the grab-and-play mapping could be a fun first musical experience for these young people. Supervised learning could later allow them to more deeply explore customisation.

These observations suggest improvements for our future workshops, in which we plan to experiment with other musical activities, and to test future grab-and-play implementations. Other input devices (such as joysticks, Wiimotes, or dance maps) as well as other output programs (such as instrument-specific samples or visual outputs) could be used to design instruments that are even better customised to each participant. Further, the social aspect of collective musical practices could be investigated through grab-and-play mapping, for example by having different young people exchanging their newly-created models, or more simply by having teachers sing with young people's sonic outputs.

5. INTERVIEWS WITH COMPOSERS AND PERFORMERS

We report on interviews held with three professional computer musicians to analyze how our grab-and-play approach could influence their music practice and/or composition processes.

5.1 Interview setup

5.1.1 Participants

We held individual interviews with three professional computer musicians. All three were composers and performers, as well as active in teaching computer music at university level. One reported previous experience with the Wekinator. We hoped to gather feedback to better understand how our grab-and-play approach could support embodied exploration processes and rapid mapping generation. We also aimed to collect information on ways to improve our first implementation. For instance, we wondered how much control the random generation method would leave to composers.

5.1.2 Structure

Each interview was a 30-minute exchange in which experimentation alternated with semi-structured interview questions. The musician was presented with a one-stringed GameTrak which allows the sensing of a user's 3D hand position, while the first author controlled the computer GUI and led the interview. Experimentation started with our grab-and-play paradigm, spanning regression and classification algorithms; it ended with the original supervised learning setup, using the same regression and classification algorithms. When first trying the grab-and-play setup, composers were not told about its implementation: they thus had no presuppositions when experimenting with it. They were asked about their playing strategies and how they thought it was working. Then, they were asked about

ways they could imagine improving this grab-and-play approach. Finally, they used the original Wekinator supervised learning setup, allowing them to experiment and compare the two approaches. For regression, we used a digital synthesis instrument based on similarities between physical models of the flute and electric guitar [14], potentially allowing for vast sound space exploration. Experimentation with classification relied on the sample trigger we used in the previous user study.

5.2 Observational study

5.2.1 Grab-and-play setup

The exploratory aspect of our grab-and-play approach was praised by the three composers. One of them described the system as *"kind of an enigma to solve"*, and was interested in the fact that *"it kind of challenges you: you have to explore a bit, and try to understand the criteria, or how to deal with these criteria"* to perform with it. Also, the possibility of rapidly prototyping a new instrument allowed them to experiment with very different gestural and sonic interactions. Using the same recorded gestural stream to build two instruments, one composer reported when comparing their playing that *"[he doesn't] feel any consistency between them in terms of gesture and sound: they felt like completely different mappings"*, saying he could explore them *"endlessly"*.

Different strategies were adopted to exploit the system's capabilities. One composer first spent time exploring the sonic parameter space, then tried to regain control and to replicate certain sounds. He then decided to reduce the space he was exploring by moving the controller in a given plane rather than in 3D, allowing him to learn certain gesture-to-sound relationships in a *"pleasant"* way. In this sense, one composer reported he could eventually learn how to play such an instrument. After having been told gestural data was randomly selected, one composer tried to exploit this aspect by spending more time in certain locations in his gestural space to increase the likelihood of their inclusion in the mapping. He indicated he was interested in *"playing more"* with this exploit.

The random selection also had some weaknesses: for example, a composer reported he had too little gestural space to explore between two interesting sounds in a given mapping. Another composer said he would require more control over the selection of sound parameters while agreeing that randomly selecting could *"definitely"* go with his vision of composing (*"the embodiment of being able to control the sound with enough level of control, regardless of what the movement is"*). Ways to modify a given mapping would be required as an improvement (this is discussed in section 5.2.3).

5.2.2 Original Wekinator setup

When testing the original Wekinator setup, one composer underlined its effect on his expectation of how a given instrument would work: *"it sets up all the run expectations, and it also affects the way I play it, because now I start to remember these poses, rather than just exploring in an open-ended way"*. Choosing gestures when building a mapping can thus be a responsibility composers want to avoid when creating meaning through sound. In this sense, a composer even mentioned that he *"never care[s] about gesture"* in composition, rather seeing these gestures as movements that are related to his own instrument practice: *"actually, what I care about is the exploration process afterwards"*.

On the other hand, one composer liked the fact that he could immediately replicate a given sound as he *"kind of see[s] what's being mapped there"*. He enjoyed the idea of spending less time on exploration and having more control, as *"in some kind of performance, you want to be very meticulous"*. Comparing the grab-and-play and original Wekinator setups, composers seemed to agree that *"both are useful"*, depending on what they would want to achieve. *"If you set up the mapping yourself, and the system yourself, you have more control, but then again maybe it's too predictable"*, one composer summed up.

5.2.3 Suggestions for improvement

Talking about ways to improve such setups, one composer evoked the idea of *"a hybrid approach"*, where one could record specific gesture-sound relationships and add some randomness in between: *"some points could be manually controlled, and some points automatically"*. This would be a way to address the previously-mentioned trade-off between control and exploration: one could then explore and discover the control space during performance, while having access to predetermined gesture-sound relationships in the mapping.

The random selection was praised for its rapidity in prototyping and experimenting, as for *"most trainings, actually, you're not really so concerned about the specific thing that's done: you just want stuff mapped out"*. However, composers would like to have a bit more control over both gesture and sound when building such a mapping. In this sense, one could imagine clever ways to select gestural and sound parameters that would still enable rapid instrument prototyping. Going further, one composer suggested incorporating the design process within the performance. Instead of being *"a static thing"*, the design process would become a real-time evolution of one's control space (*"me creating the control space in real-time"*). For example, such a performance could entail repeating the same gesture to tell the machine to add new sounds to this gesture. This idea is reminiscent of Fiebrink's play-along mapping approach [15].

Finally, one composer noticed the difficulty in editing a newly-generated mapping: *"It's really frustrating when you're working musically because you just want to tweak that thing, and then the whole thing blows up"*. One could edit the training data, or, as the composer suggested, *"regression is just a geometry, so why can't we just start stretching things and manipulate them?"* Designing a user interface that allows the intuitive modification of an N-dimensional geometry would be necessary; however, this goes beyond the scope of our grab-and-play mapping paradigm.

5.2.4 General discussion

These individual interviews have clarified what kind of compositional processes could be allowed with our grab-and-play approach. Composers' opinions globally corresponded to our intuitions about the discovery and exploration processes encouraged by our first implementation of the tool.

As mentioned by one composer, such a random process may be used when starting a piece, as a way to let new ideas emerge, then opening up a reflection on how to use them: quoting him, *"all these mapping processes are about making decisions that are rational: it's just building blocks. Then, musical decisions come as you actually walk through them..."*

Other implementations of our grab-and-play paradigm may also support composers' needs (see Figure 2). For example, clustering gestural data could meet composers' need for control over their gestural space in relation to sound, while allowing rapid prototyping. This setup is already implemented but not yet tested. Also, most composers wanted to have more control over the choice of sounds: in future work, we would like to allow a user to choose output labels by selecting high-level perceptual characteristics of a synthesis engine's sound space. Finally, hybridizing grab-and-plau mapping with the original supervised learning setup could be a way to encourage discovery while allowing customization. We plan to experiment with each of these implementations in the near future.

6. CONCLUSIONS AND FUTURE WORK

We presented a first implementation of our "grab-and-play" approach to mapping that allows the prototyping of digital musical instruments. We reported on a first workshop with disabled young people, suggesting that the tool could be useful in the context of musical inclusion. The rapid prototyping of adapted musical interfaces allowed youth with less concentration to instantaneously take part in musical activities, while those with more concentration were curious about both grab-and-play and supervised learning setups, notably enjoying the customization of the latter. We also reported on interviews with three composers and performers, suggesting that the tool could encourage the realisation of new musical outcomes. Each of them valued the grab-and-play approach for embodied musical exploration, and underlined the balance between discovery and control that such a paradigm could support. Their feedback allowed us to imagine future improvements to the current implementation. More generally, the grab-and-play's simple yet expressive framework reflects our wish to get more people progressively included in modern musical activities, and in a broader sense, to have them create new technologies more easily.

In the next two years we will develop our contribution to musical inclusion through workshops and prototypes that will implement more engaging musical activities that are specifically adapted to a participant's abilities. We are also currently implementing more sophisticated ways to select gestural inputs and sound outputs. Using unsupervised learning algorithms to extract relevant clusters from the recorded gestural stream could be a possibility. Another possibility would be to generate input data that are more equally spread through the space delimited by user's gestural extrema. The choice of output labels could also be informed by the relationship between synthesis parameters and higher-level perceptual characteristics, enabling the creation of instruments capable of accessing a desired perceptual sound space. Hybrid approaches mixing grab-and-play mapping with user-provided pairs of inputs and outputs could also be a way to encourage exploration while allowing customization. More generally, we believe that having digital musical instruments generate their own gestural interactions just as they generate sounds could be an engaging conceptual framework, both scientifically and artistically, as it remains mostly unexplored in the context of computer music.

Acknowledgments

We thank Simon Steptoe and the Northampton Music and Performing Arts Trust for inviting us to join their Musical Inclusion Programme (funded by Youth Music and the Paul Hamlyn Foundation). We also thank Simon Katan, Mick Grierson and Jérémie Garcia for useful discussions and suggestions.

7. REFERENCES

[1] A. Hunt and M. M. Wanderley, "Mapping performer parameters to synthesis engines," *Organised sound*, vol. 7, no. 02, pp. 97–108, 2002.

[2] R. Fiebrink, D. Trueman, C. Britt, M. Nagai, K. Kaczmarek, M. Early, M. Daniel, A. Hege, and P. Cook, "Toward understanding human-computer interaction in composing the instrument," in *Proc. International Computer Music Conference*, 2010.

[3] M. Lee, A. Freed, and D. Wessel, "Real time neural network processing of gestural and acoustic signals," in *Proc. International Computer Music Conference*, 1991.

[4] R. A. Fiebrink, "Real-time human interaction with supervised learning algorithms for music composition and performance," Ph.D. dissertation, Princeton University, 2011.

[5] B. Caramiaux and A. Tanaka, "Machine learning of musical gestures," in *Proc. International Conference on New Interfaces for Musical Expression*, 2013.

[6] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana, "Continuous realtime gesture following and recognition," in *Gesture in embodied communication and human-computer interaction*. Springer, 2010, pp. 73–84.

[7] J. Francoise, "Motion-sound mapping by demonstration," Ph.D. dissertation, Université Pierre et Marie Curie, 2015.

[8] M. Leman, *Embodied music cognition and mediation technology*. MIT Press, 2008.

[9] R. I. Godøy and M. Leman, *Musical gestures: Sound, movement, and meaning*. Routledge, 2010.

[10] B. Caramiaux, F. Bevilacqua, T. Bianco, N. Schnell, O. Houix, and P. Susini, "The role of sound source perception in gestural sound description," *ACM Transactions on Applied Perception (TAP)*, vol. 11, no. 1, p. 1, 2014.

[11] H. Scurto, G. Lemaitre, J. Françoise, F. Voisin, F. Bevilacqua, and P. Susini, "Combining gestures and vocalizations to imitate sounds," *Journal of the Acoustical Society of America*, vol. 138, no. 3, pp. 1780–1780, 2015.

[12] M. Wright and A. Freed, "Open sound control: A new protocol for communicating with sound synthesizers," in *Proc. International Computer Music Conference*, 1997.

[13] S. Katan, M. Grierson, and R. Fiebrink, "Using interactive machine learning to support interface development through workshops with disabled people," in *Proc. SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 251–254.

[14] D. Trueman and R. L. DuBois, "Percolate," *URL: http://music.columbia.edu/PeRColate*, 2002.

[15] R. Fiebrink, P. R. Cook, and D. Trueman, "Play-along mapping of musical controllers," in *Proc. International Computer Music Conference*, 2009.

The problem of musical gesture continuation and a baseline system

Charles Bascou
gmem-CNCM-marseille
first.last@gmem.org

Valentin Emiya
Aix-Marseille University
CNRS UMR 7279 LIF
first.last@univ-amu.fr

Mathieu Laurière
Paris Diderot University
UPMC CNRS UMR 7598 LJLL
mlauriere@math.univ-paris-diderot.fr

ABSTRACT

While musical gestures have been mapped to control synthesizers, tracked or recognized by machines to interact with sounds or musicians, one may wish to continue them automatically, in the same style as they have been initiated by a performer. A major challenge of musical gesture continuation lies in the ability to continue any gesture, without a priori knowledge. This gesture-based sound synthesis, as opposed to model-based synthesis, would open the way for performers to explore new means of expression and to define and play with even more sound modulations at the same time.

We define this new task and address it by a baseline continuation system. It has been designed in a non-parametric way to adapt to and mimic the initiated gesture, with no information on the kind of gesture. The analysis of the resulting gestures and the concern with evaluating the task raise a number of questions and open directions to develop works on musical gesture continuation.

1. THE PROBLEM OF MUSICAL GESTURE CONTINUATION

1.1 Musical gesture

From traditional acoustic instruments to modern electronic musical interfaces, gesture has always been a central problematic in musical performance. While acoustic instruments have to be continuously excited by energy impulsed by the performer's gestures, electronic instruments produce sounds without any mechanical input energy, which can last as long as electricity flows. In such electronic instruments, gestural interfaces have been used to control sound synthesis parameters. In [1], electronic instruments are defined by two components – the gestural controller and the sound production engine – and by the mapping of parameters between them.

The electronic performer can now deal with multiple layers of sound, mixed together as tracks on traditional digital audio workstation. Even if gestural control can be at a high level in the music system architecture – *e.g.*, on a mixing desk –, we often use several electronic instruments with performance-dedicated control strategies and interfaces. As they can't necessarly be played simultaneously, here comes the idea to design a system that continues a gestural behavior, primarily inputed by the performer on a particular instrument, and then automatically continued, while the performer can focus on other instruments.

Another motivation of such systems is the ability to define complex sound parameter modulations by gesture. From very simple Low Frequency Oscillators to chaotic systems, modulation methods are often parametric. One can use simple periodic/stochastic function or linear combination of these functions. This leads to very complex and rich results in terms of dynamics and movements but with a real pain on tweaking parameters. Indeed, these systems propose a lot of parameters, with complex interactions, making them really difficult to control intuitively. The idea to define modulation by gesture comes quite straightforward. Such a data-driven approach, as opposed to model-based parametric systems, leads to a system that could analyze an input gesture by means of its temporal and spatial characteristics, and then continue it *à la manière de*.

1.2 Problem characterization

Let us imagine an electronic instrument controlled by a tactile tablet. Consider gestures as isolated 2D strokes related to the contact of a finger on that tablet. An example of such a gesture is represented in black in Figure 1, together with a possible continuation of this gesture, in gray. This setting will be used throughout the proposed study and extensions to other settings will be discussed.

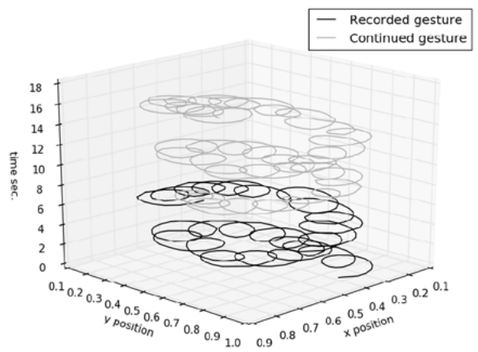


Figure 1. Example of a continuation (gray) of a performed 2D gesture (black). Space is on the x and y axes while time is on the z axis.

This work was partially supported by GdR ISIS project Progest and by the French National Research Agency (ANR), with project code MAD ANR-14-CE27-0002, Inpainting of Missing Audio Data. The authors would like to thank Pr. Franois Denis (LIF) for his fruitful scientific inputs.

We can formalize the problem with a traditional machine learning scheme composed of a learning phase (user performing the gesture) followed by a predicting phase (machine continuing the gesture). A gesture is a sequence of positions $\mathbf{p}(t) \in \mathbb{R}^2$ in cartesian coordinates for a single finger recorded at N discrete times t with $1 \leq t \leq N$. The goal of gesture continuation is to extrapolate a given gesture by generating the positions after the end of the available recording, *i.e.*, to estimate $\mathbf{p}(t)$ at times $t > N$.

The key issue is to study to which extent one may continue any gesture, with no *a priori* knowledge on the properties of the gesture. In particular, we want to avoid any categorization of gestures that would lead to, *e.g.*, parametric models of specific gestures. For instance, we are not interested in tracking periodic gestures to generate perfect loops, since the musical result would be too simplistic and is already well-used by say live-looping techniques, or to have a predefined list of allowed gesture patterns, which would dramatically reduce the performer's freedom. On the contrary, one may want, for instance: to capture the variability in the gesture of the performer, including when it is periodic ; to be able to continue aperiodic gestures that look like random walks; to reproduce the main characteristics of the gesture, including, at the same time, oscillating or random components – even if such structures do not appear in the sequence of positions, but in the velocity space for instance. Consequently, musical gesture continuation is not a well-posed problem. This important aspect should be considered when designing continuation systems as well as evaluation frameworks in order to keep in mind the ultimate goal – processing any gesture – and to avoid excessive simplification of the problem.

1.3 Related tasks

Part of the problem of musical gesture continuation is obviously related with the sound generation and mapping strategies involved in the electronic instrument in use. Indeed, gestures are completely dependent on the audio feedback, involving the need to study relations between sound and movement as in [2]. We chose for now to use a fixed reference electronic instrument, to work in the gesture domain only, and to tackle this question in future works.

Gesture continuation differs from other tasks that involve either gestures or continuation in music. In the gesture analysis field, gesture recognition [3, 4, 5] relies on reference gestures that are available beforehand and may be used to follow and align various media (sound, musical score, video) in live performance. Such reference gestures are not available in the generic gesture continuation problem. In [6, 7], the authors propose a system that can continue musical phrases and thus improvise in the same style. It works at a symbolic level (discrete segmented notes or sounds) and its application to continuous data (ie. gesture time series) is not straightforward.

1.4 Outline

This paper is organized as follows. In section 2, we propose a baseline system that has been designed to continue any arbitrary gesture, in the spirit of the open problem described above. A large place is dedicated to the evaluation of the results in section 3, including questions related to the evaluation methodology. We finally discuss a number of directions for this new problem in section 4.

2. A BASELINE SYSTEM BASED ON K-NEAREST NEIGHBORS REGRESSION

The proposed baseline system for musical gesture continuation is based on a simple regression scheme, as presented in section 2.1. In order to be able to continue any gesture, the system relies on the design of feature vectors discussed in section 2.2. The choice of the prediction function is finally detailed in section 2.3.

2.1 Overview and general algorithm

The proposed approach relies on the ability, at each time $t > N$, to generate the move $\boldsymbol{\delta}(t) \in \mathbb{R}^2$ from the current position $\mathbf{p}(t)$ to obtain the next one as $\mathbf{p}(t+1) = \mathbf{p}(t) + \boldsymbol{\delta}(t)$, by considering the past and current positions

$$\mathbf{x}(t) \triangleq \left[\mathbf{p}(t-\tau^o), \ldots, \mathbf{p}(t)\right] \tag{1}$$

where τ^o is a predefined memory length.

The proposed system is depicted in Algorithm 1. It relies on learning a prediction function (lines 1-7) which is then used to predict the moves at times $t \geq N$ (lines 8-13).

At each time t during both learning and prediction, a feature vector $\mathbf{v}(t)$ is computed from the current data point $\mathbf{x}(t)$ (lines 2-3 and 9-10).

The recorded gesture provides examples $(\mathbf{v}(t), \boldsymbol{\delta}(t))$ of the mappings between a feature vector $\mathbf{v}(t)$ and a subsequent move $\boldsymbol{\delta}(t)$ for $t \in \{1+\tau^o, \ldots, N-1\}$. Such a training set $\mathcal{S}$ is built at line 6. The prediction function $f_{\mathcal{S}}$ is obtained from $\mathcal{S}$ at line 7 by a supervised learning step. Once the prediction function is learned, gesture continuation is obtained in an iterative way for times $t \geq N$, by applying $f_{\mathcal{S}}$ to the feature vector $\mathbf{v}(t)$ in order to obtain the subsequent move (line 11) and the next position $\mathbf{p}(t+1)$ (line 12).

2.2 Feature extraction

In order to be as generic as possible, we consider simple features based on position, speed and acceleration along the gesture. Two options are proposed, in sections 2.2.1 and 2.2.2.

2.2.1 Instantaneous features

The simplest option consists in setting the characteristic memory length τ^o to 2 so that the point $\mathbf{x}(t) = \begin{bmatrix}\mathbf{p}(t-2)\\ \mathbf{p}(t-1)\\ \mathbf{p}(t)\end{bmatrix}$ considered at time t is composed of the last three positions. The feature vector is then defined as

$$\mathbf{v}^{\text{inst}}(t) \triangleq \begin{bmatrix}\mathbf{p}(t)\\ \mathbf{p}(t) - \mathbf{p}(t-1)\\ \mathbf{p}(t) - 2\mathbf{p}(t-1) + \mathbf{p}(t-2)\end{bmatrix} \in \mathbb{R}^6,$$

by concatenating the current position $\mathbf{p}(t)$, the instantaneous speed $\mathbf{p}(t) - \mathbf{p}(t-1)$ and the instantaneous acceleration $\mathbf{p}(t) - 2\mathbf{p}(t-1) + \mathbf{p}(t-2)$ computed in a causal way from point $\mathbf{x}(t)$.

Algorithm 1 Gesture continuation

Input(s):
recorded gesture $(\mathbf{p}(t))_{1\leq t\leq N}$
prediction length L
Output(s):
predicted gesture $(\mathbf{p}(t))_{N+1\leq t\leq N+L}$

Supervised learning on recorded gesture
1: **for** $t \in \{1+\tau^o, \ldots, N-1\}$ **do**
2: build point $\mathbf{x}(t) \leftarrow (\mathbf{p}(t-\tau^o), \ldots, \mathbf{p}(t))$
3: build feature vector $\mathbf{v}(t)$ from $\mathbf{x}(t)$
4: set move $\boldsymbol{\delta}(t) \leftarrow \mathbf{p}(t+1) - \mathbf{p}(t)$
5: **end for**
6: build training set $\mathcal{S} \leftarrow \{(\mathbf{v}(t), \boldsymbol{\delta}(t))\}_{t=1+\tau^o}^{N-1}$
7: learn regression function $f_{\mathcal{S}}$ from $\mathcal{S}$

Prediction
8: **for** $t \in \{N, \ldots, N+L-1\}$ **do**
9: build point $\mathbf{x}(t) \leftarrow (\mathbf{p}(t-\tau^o), \ldots, \mathbf{p}(t))$
10: build feature vector $\mathbf{v}(t)$ from $\mathbf{x}(t)$
11: estimate move $\boldsymbol{\delta}(t) \leftarrow f_{\mathcal{S}}(\mathbf{v}(t))$
12: set next position: $\mathbf{p}(t+1) \leftarrow \mathbf{p}(t) + \boldsymbol{\delta}(t)$
13: **end for**

2.2.2 *Finite-memory features*

Instanteous features may provide insufficient information to predict the next position, as it will be experimentally demonstrated (see section 3). An alternative choice is proposed by extending the memory length τ^o and by considering information at J different past lags t_j in the range $\{0, \ldots, \tau^o - 2\}$. We define the finite-memory feature vector

$$\mathbf{v}^{\text{mem}}(t) \triangleq \left[\mathbf{v}^{\text{inst}}(t-t_j)\right]_{j=0}^{J} \quad (2)$$

as the concatenation of several instantaneous feature vectors $\mathbf{v}^{\text{inst}}(t-t_j)$ taken at past times $t-t_j$ within the finite memory extension. In order to exploit the available information while limiting the feature vector size, the finite-memory data is sampled on a logarithmic scale by setting:

$$J \triangleq \lfloor \log_2(\tau^o - 2) + 1 \rfloor$$
$$t_0 \triangleq 0 \text{ and } t_j \triangleq 2^{j-1} \text{ for } 1 \leq j \leq J$$

where $\lfloor . \rfloor$ denote the floor function.

2.3 Prediction function

The desired prediction function maps a feature vector to a move in $\mathbb{R}^2$. Learning such a function from a training set $\mathcal{S}$ is a regression problem for which many well-known solutions exist, from the most elementary ones – *e.g.*, K-nearest neighbors regression, kernel ridge regression, support vector regression – to the most advanced ones – *e.g.*, based on deep neural networks. Since comparing all those approaches is not in the scope of this paper and since we target real-time learning and prediction, we use one of the simplest ones. The resulting system may serve as a baseline for future works and any other regression method may replace the proposed one in a straightforward way.

We use a K-nearest neighbors (KNN) regression approach based on a predefined number of neighbors K and the euclidian distance as the metric d in the feature space. Learning the regression function $f_{\mathcal{S}}$ from the training set composed of labeled examples $\mathcal{S} = \{(\mathbf{v}(t), \boldsymbol{\delta}(t))\}_{t=1+\tau^o}^{N-1}$ simply consists in storing $\mathcal{S}$ for further neighbor search. The KNN regression function is given by the Algorithm 2 and is used at line 11 in Algorithm 1. It first selects indices $(k_1, \ldots, k_K)$ of the K nearest neighbors of the current feature among the feature vectors $(\mathbf{v}_1, \ldots, \mathbf{v}_{N_{\mathcal{S}}})$ of the training set; and then define the predicted move $\boldsymbol{\delta}$ as the average of the related moves $(\boldsymbol{\delta}_{k_1}, \ldots, \boldsymbol{\delta}_{k_K})$.

Algorithm 2 KNN regression function $(f_{\mathcal{S}}(\mathbf{v}))$

Input(s):
training set $\mathcal{S} = \{(\mathbf{v}_k, \boldsymbol{\delta}_k)\}_{k=1}^{N_{\mathcal{S}}}$ with size $N_{\mathcal{S}}$
feature vector $\mathbf{v}$
number of neighbors K
distance d in feature space
Output(s): move $\boldsymbol{\delta} \in \mathbb{R}^2$

1: find the K-nearest neighbors of $\mathbf{v}$ in $\mathcal{S}$ as

$$\{k_1, \ldots, k_K\} \leftarrow \underset{\{k_1,\ldots,k_K\}\subset\{1,\ldots,N_{\mathcal{S}}\}}{\arg\min} \sum_{i=1}^{K} d(\mathbf{v}_{k_i}, \mathbf{v})$$

2: average moves of selected neighbors

$$\boldsymbol{\delta} \leftarrow \frac{1}{K}\sum_{i=1}^{K} \boldsymbol{\delta}_{k_i}$$

3. EXPERIMENTS AND EVALUATION

Designing an evaluation framework for gesture continuation is a complex topic for which we first raise a number of issues.

First of all, one should keep in mind the main goal – continuing subjective gestures with no a priori contents –, even as an unreachable objective; in particular, one should find how this can properly make part of an evaluation.

The space of musical gestures with no a priori may have a complexity much larger than what can be represented in some training and testing sets, and gathering a representative and statistically-consistent set of gestures may be a vain wish. This issue will impact the classical use of a training set (*e.g.*, for tuning parameters by cross-validation) as well as the validity of performance assessment on a testing set.

In terms of performance measure, one may not hope for a well-defined global score available beforehand. One may even consider it is a too challenging task to evaluate the quality of a predicted musical gesture by integrating complex aspects like: the intention of the gesture author and his or her subjective idea of what is a good continuation; differences in the audio rendering of various possible gesture continuations belonging to some kind of equivalence class, beyond a singular groundtruth gesture. In such conditions, one may characterize the main evaluation objective by the following two uncommon statements: the evaluation criteria may vary from one gesture to another; the evaluation criteria may be established by the performer at the time the gesture is generated so that it may not be known at the time a continuation system is designed. In such context, one may develop an *a posteriori* multi-criteria evaluation and consider it as a principal evaluation method while usual *a priori* evaluation criteria may play a subordinate role.

Hence, the design of an evaluation framework for musical gesture continuation is an important challenge. We propose a substantial evaluation section which may be thought of as a first step in that direction.

Our experimental protocol is based on a corpus of recorded gesture to be continued. This set has been made with pluricity and maximum variability in mind, combining strong periodic forms to pseudo-random paths. A first set of 12 gestures was composed of basic geometrical shapes like circles, triangles, oscillations. A second set of 18 gestures has been made by a musician who was asked to specifically focus on the sound result. Gestures are sampled from a 2D tactile tablet at a rate of 60 points per seconds. The gesture data is sent to a custom sound synthesis software and stored as textfiles one line per point. Their length varies between 9.3s and 29.6s, the average being 18.8s.

We first analyze isolated gestures to provide a short *a posteriori* analysis in section 3.1. We then define an objective measure for prediction accuracy and apply it to evaluate the effect of finite-memory features in section 3.2. Finally, we propose a multicriteria evaluation framework in order to help the analysis of gesture continuation by pointing out a set of key evaluation criteria.

3.1 Three notable continuation examples

Figure 2 shows, for three gestures, their continuations by the proposed system with instantenous features only ($J = 0$) and with finite-memory features ($J = 7$, *i.e.*, about 2 seconds). The number of neighbors is fixed to $K = 5$. Those example have been chosen to illustrate the ability of the proposed approach to continue any gesture, as well as its limitations.

In the case of a gesture with a strong stochastic component (example on the left), both continuations show some ability to generate a similar stochastic behavior. It seems that finite-memory features help to reproduce a large variability including spatial spread and temporal cues. One may notice that short patterns of the initial gesture are locally reproduced. However, the system does not seem to be trapped in a loop, which would have had a strong negative impact on the perceived musical forms. From this point of view, the evaluation on random-like gestures may be compared to the topic of pseudo-random numbers generation where one tries to avoid any repetitions or period in the sequence of generated numbers.

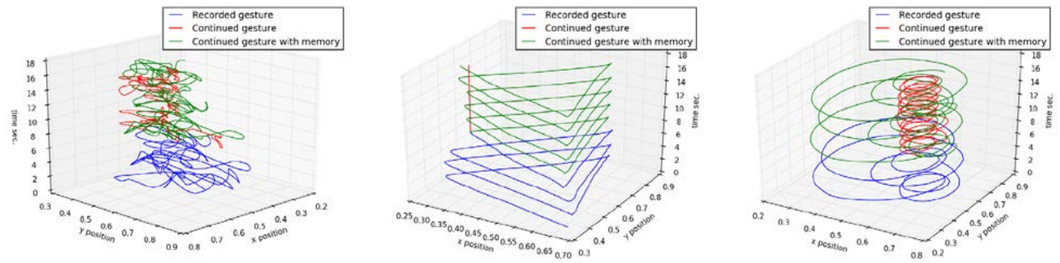


Figure 2. Initial gesture and two possible continuations for : a random-walk-like gesture (left); a triangle-shape gesture (middle) a periodic motion that alternates two small circles and a large one (right). The red and green gestures have been generated with the systems with instantaneous features only ($J = 0$) and with finite-memory ($J = 7$), respectively.

The continuation of quasi-periodic gestures is illustrated in the other two examples. A recurrent failure case of the system with instantaneous features only is illustrated by the triangle-shaped gesture where speed is constant on edges and is null on corners (the user deliberately stops its movement for a while at each corner). The system is trapped in a corner since the K nearest neighbors have null-speed and the related move is also null. Finite-memory features provide information from the current point history to avoid such problems, as long as that the memory length is greater than the duration of stops. In the obtained (green) continuation, one may observe that the system succeeds in generating triangles, and that they present a variability similar to that of the original gesture.

Another common challenging situation is crosspoints in periodic motions, as illustrated in the third example. The gesture is a repetition of one large circle and two small circles successively. All three circles are tangent at their junction point, which generates an ambiguity since at that point, position, speed and acceleration are similar. Hence, at that position, the system with instantaneous features only is not able to determine whether it should enter a large or a small circle and gets trapped into the small circles here. On the contrary, the system with finite-memory features uses history information and is able to generate an alternation of one large circle and two small circles.

From these examples, one may note how gesture-dependent the evaluation is. Indeed for each example, specific evaluation criteria have been commented on, based on the property of the gesture as well as on the behavior of the system. This shows the importance of *a posteriori* evaluation. In a more synthetic way, one may also conclude from these examples that the proposed system is able to continue gestures of very different natures and that finite-memory features are useful to avoid typical failures. The subsequent sections will provide an extensive evaluation on this topic.

3.2 Prediction accuracy for various memory sizes

The memory size have a dramatic effect on the prediction results: one may wonder how large it should be set and how the system behaves when it varies. We propose to introduce and use an objective measure to assess the qual-

ity of the prediction at various horizons after the last point used to learn the gesture. This measure is subsequently used to analyze the prediction accuracy as a function of the memory size.

Let us consider a recorded gesture of total length $\widetilde{N}$: for clarity, $\widetilde{\mathbf{p}}(t)$ denote the recorded position for $1 \leq t \leq \widetilde{N}$. We denote by $N_0 < \widetilde{N}$ a minimum size considered for training. For a training size N such that $N_0 \leq N < \widetilde{N}$, the system is trained on the first N positions only and for $t > N$, $\widehat{\mathbf{p}}^{(N)}(t)$ denotes the position predicted by this system at time t. In such conditions, for any $n \leq \widetilde{N} - N$, position $\widehat{\mathbf{p}}^{(N)}(N+n)$ is the position predicted at a horizon n after the last position N known by the system. We define the mean prediction error at horizon n, $1 \leq n \leq \widetilde{N} - N_0$, by

$$\epsilon(n) \triangleq \frac{\sum_{N=N_0}^{\widetilde{N}-n} \left\| \widehat{\mathbf{p}}^{(N)}(N+n) - \widetilde{\mathbf{p}}(N+n) \right\|_2}{\widetilde{N} - n - N_0 + 1}. \quad (3)$$

In other words, for a fixed horizon n, $\epsilon(n)$ is the prediction error averaged among the predictions at horizon n obtained by training the system on different sizes N of training set, using the same recorded gesture.

Figure 3 shows the mean error averaged over all the gestures we considered, when fixing the number of neighbors to $K = 5$ and and the minimum training size to $N_0 = \frac{2}{3}\widetilde{N}$ (first two thirds of each gesture). The error increases with the horizon, since it is harder to make an accurate prediction when the horizon is large. Each curve can be split into two parts. During a first phase (memory size below 0.5 second), increasing the memory helps decreasing the error significantly. However, increasing the memory size beyond 0.5 second does not improve the prediction and sometimes drives up the error. These two trends (decreasing and then increasing) are found in most of the examples we considered, with different optimal memory sizes from one gesture to the other, and show that the proposed system has a limited capacity to learn from past points.

One may also note that this evaluation measure is not well suited for some gestures. For instance, if a gesture is made up of randomness, all possible realizations of this randomness are satisfying ways to continue it. As a consequence, a valid extrapolated gesture might be very far from the actual continuation made by the user. In this perspective, it appears useful to introduce other evaluation criteria.

3.3 Multicriteria evaluation

Evaluation may be thought within a multicriteria framework, relying on multiple evidence, by extending the use of objective performance measures. Since the criteria are not combined into a single score, this methodology is not dedicated to learn parameters or to rank concurrent systems. Generic multiple criteria may be used as a set of objective features that are automatically generated to help human interpretation or analysis.

We propose a set a evaluation criteria in order to compare a continued gesture $\widehat{\mathbf{p}}$ and a groundtruth continuation $\widetilde{\mathbf{p}}$. The experimental setting consists in splitting each gesture with a ratio $(2/3, 1/3)$ so that the first part is used for learning and is continued by the system to obtain $\widetilde{\mathbf{p}}$ while the second part is taken as the groundtruth $\widetilde{\mathbf{p}}$ for performance assessment ($\widehat{\mathbf{p}}$ and $\widetilde{\mathbf{p}}$ having the same length). The

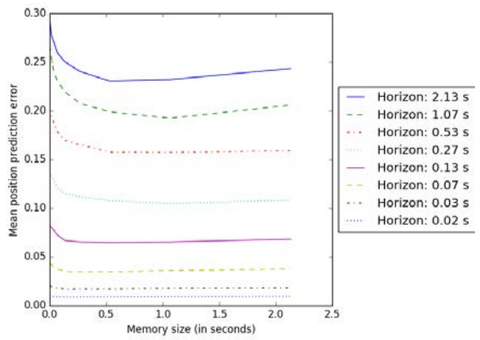


Figure 3. Mean prediction error averaged over all gestures for several prediction horizons n, as a function of the memory size J (n and J have been converted in seconds).

proposed criteria are based on instantaneous features in the continued gesture: position, speed and acceleration vectors, as well as their norms and angles. In each of those 9 possible cases, the feature of interest is extracted from the continued gesture $\widehat{\mathbf{p}}$ and from the groundtruth $\widetilde{\mathbf{p}}$ at each available sample time, resulting in two feature vectors to be compared.

A first family of criteria aims at analyzing the distribution of the instantaneous features. Distributions are considered by building histograms from the coefficients of feature vectors. For each feature, we compare the histogram for the continued gesture and that of the grountruth using a simple histogram difference measure, both histograms being computed on a common support with size $N_b = 25$ bins. Results are represented in the top part of Figure 4. In order to separate the gesture trajectory from its dynamics, a second family of criteria is proposed, based on dynamic time warping (DTW). DTW is used to align the continued gesture and the groundtruth, which cancels the effect of possible time stretching : the obtained distance measure quantifies only the difference in the trajectories, evaluating spatial cues only. Results are denoted by DTW/position in the middle part of Figure 4. As a possible extension, we also represent the DTW computed on the vectors of instantaneous speed, acceleration and speed norm instead of positions. Finally, since many gesture have one or several oscillating components – sometimes in position, speed, acceleration, and so on –, we also computed the Fourier transform of feature vectors. For each feature, spectra from the continued gesture and from the groundtruth are compared using the log-spectral distance and results are presented in the bottom part of Figure 4.

Results shown in Figure 4 confirm the advantage of finite-memory features in the proposed continuation system, since almost all criteria are improved on average. This multicriteria framework may also be used to detect gestures that are not well continued – *e.g.*, by automatically selecting gestures that are in the higher quartile – in order to draw a detailed analysis. As not all criteria are of interest for a given gesture, the performer may select them from this full dashboard, on a gesture-dependent basis, adopting an *a posteriori* evaluation.

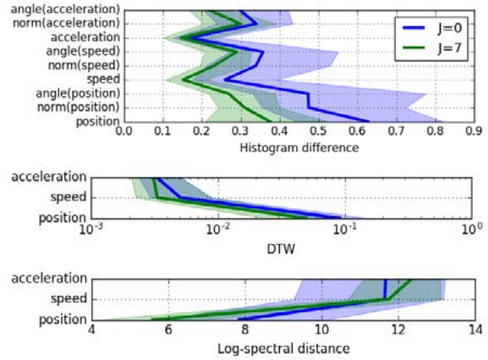


Figure 4. Multicriteria evaluation of the proposed system with instantaneous features only ($J = 0$, $K = 5$) and finite-memory features ($J = 7$, $K = 5$). Plain curves are median values among all gestures, with interquartile ranges as shaded areas. Various families of criteria are represented from top to bottom.

4. CONCLUSION AND PERSPECTIVES

We would like the main conclusion of this paper to be that the problem of musical gesture continuation, despite its vague definition, is not a vain or absurd task. To support this conclusion, we have shown that a system based basic features and KNN regression is able to continue any arbitrary gesture in an automatic way. We have also proposed the guidelines for an evaluation framework, including some particular considerations on specific gestures (null velocity issues, periodic and random components), a prediction accuracy measure and a large set of multicriteria objective measures that may be used in an *a priori* evaluation setting as well as for *a posteriori* evaluation. Those elements form preliminary contributions for works on musical gesture continuation, with several open directions.

The problem setting and the evaluation framework should go beyond the proposed ideas. 2D gestures may include multiple strokes generated simultaneously (*e.g.*, by several fingers) and sequentially (with arbitrary stops between strokes). They may also be extended to 3D gestures. The set of evaluation criteria may be completed by other features and comparison measure computed on the gesture itself, as well as criteria in the audio domain. This may also be the opportunity to analyze the relation between gesture and audio domains. Finally, subjective evaluation should also be considered and would first require the design of dedicated test protocols.

The proposed system for gesture continuation may be extended in some interesting directions. As shown in this paper, a significant improvement results from the extension of instantaneous features to finite-memory features. Adding more features may be even more useful to capture the right information, using feature selection method at training time. As a more fundamental issue, one may design or learn an appropriate distance in the feature domain while features are numerous and of different natures. We think that metric learning approaches would play an important role in order to have continuation systems that adapt to each gesture. One may also explore the wide range of possible non-parametric prediction functions. For instance, hidden Markov models may be successful to model the time dependencies as well as to control variations from the reference gesture as in [8].

Is the sky the limit? In many aspects, the problem of musical gesture continuation raises important questions about how to go beyond the limits we usually set for prediction tasks: how to deal with the dilemma of characterizing musical gestures with no *a priori*? How to address ill-posed problems as such? How to design systems when evaluation criteria are not known? Eventually, would such works be of interest to revisit conclusions from well-established tasks, as they may be questioned in [9]?

5. REFERENCES

[1] M. Wanderley and P. Depalle, "Gestural control of sound synthesis," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 632–644, Apr 2004.

[2] A. Hunt, M. M. Wanderley, and M. Paradis, "The Importance of Parameter Mapping in Electronic Instrument Design," *Journal of New Music Research*, vol. 32, no. 4, pp. 429–440, 2003.

[3] G. Lucchese, M. Field, J. Ho, R. Gutierrez-Osuna, and T. Hammond, "GestureCommander: continuous touch-based gesture prediction," in *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2012.

[4] M. Takahashi, K. Irie, K. Terabayashi, and K. Umeda, "Gesture recognition based on the detection of periodic motion," in *Int. Symp. on Optomechatronic Technologies (ISOT)*. IEEE, 2010, pp. 1–6.

[5] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana, "Continuous realtime gesture following and recognition," in *Gesture in Embodied Communication and Human-Computer Interaction*, ser. LNCS. Springer Verlag, 2010, vol. 5934, pp. 73–84.

[6] F. Pachet, "The Continuator: Musical Interaction with Style," *Journal of New Music Research*, vol. 32, no. 3, pp. 333–341, 2003.

[7] G. Assayag and S. Dubnov, "Using Factor Oracles for machine Improvisation," *Soft Computing*, vol. 8, no. 9, Sep. 2004.

[8] B. Caramiaux, N. Montecchio, A. Tanaka, and F. Bevilacqua, "Adaptive Gesture Recognition with Variation Estimation for Interactive Systems," *ACM Trans. Interact. Intell. Syst.*, vol. 4, no. 4, pp. 18:1–18:34, Dec. 2014.

[9] B. L. Sturm, "A Simple Method to Determine if a Music Information Retrieval System is a Horse," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1636–1644, Oct 2014.

RECONSTRUCTING ANTHÈMES 2: ADDRESSING THE PERFORMABILITY OF LIVE-ELECTRONIC MUSIC

Laurens van der Wee, MMus
Independent Artist
HKU School of Music and Technology graduate
't Goy, The Netherlands
l.vanderwee@gmail.com

Roel van Doorn, MA
Independent Artist
HKU School of Music and Technology graduate
Rotterdam, The Netherlands
roelvandoorn@gmail.com

Jos Zwaanenburg
Conservatory of Amsterdam
Amsterdam, The Netherlands
zwaanenburg@open.net

ABSTRACT

This paper reports on the reconstruction from the score of the software for Pierre Boulez' Anthèmes 2*, for violin and live-electronics. This increasingly popular piece, judged by the number of performances in recent years, has been rebuilt from scratch for the very first time. We will put this work into context, give a short description of the composition's electro-acoustic system, describe our approach and take a look into the future.*

1. INTRODUCTION

If one is interested in creating live-electronic music compositions that can be performed in the future, technological issues arise. Especially when using closed source applications, there is no guarantee whatsoever that computer programs, patches or scripts will be usable in the future. With most live-electronic music compositions however, the score is published together with a computer program.

Pierre Boulez' *Anthèmes 2* for violin and live-electronics was premiered in 1997, the score [1] was published in 2005. For its performance, software can be obtained from Ircam[1], however, the publication of Anthèmes 2 doesn't include the software, instead it is possible to create software from the score, since the electro-acoustic behaviour is transcribed as well. Building this software can be done by interpreting the score according to the instructions in the technical manual [2], included in the publication.

This was done with the aim to investigate whether this approach can serve as an alternative for the common practice of releasing a score together with a computer program, and with the performability of live-electronic music in the distant future in mind[2].

The authors built a new version of the software according to the score and report on this process in this paper.

[1] Referred to in this paper as the 'original version'.

[2] This was confirmed in an email to the authors by Andrew Gerzso. However, no such intentions are described in the score or in the technical manual.

2. RELEVANT WORK

Obsolescence of live-electronic music computer programs is a major issue in light of its performability [3]. This is obviously not only an issue in the case of Anthèmes 2. Other people have reconstructed historically relevant electronic music repertoire, not necessarily computer assisted, that may be difficult to bring back on stage otherwise. These reconstructions often involve introducing computers into a certain technological set-up. Many compositions use technology that at the time of conception was state-of-the-art, but has become rather inefficient by now. To allow these pieces to be performed more often, computer assisted versions are developed. Of course, a lot of technological issues, as well as aesthetic and conceptual ones, arise in the process. Some of this work is documented [4,5,6,7]. [8] describes the process of increasing performability by recreating works using open source technology.

Furthermore, also tape music has been reconstructed [9]. Also worth noting is a version of Boulez's *Dialogue de l'Ombre Double* for recorder (Erik Bosgraaf) and electronics (Jorrit Tamminga)[3] (the original composition is for clarinet and live-electronics).

However, none of these electro-acoustic realisations are based solely on a published score.

3. DESCRIPTION OF THE PIECE

3.1 General

Anthèmes 2 is an approximately twenty-two-minute piece for solo violin and live-electronics by Pierre Boulez, first performed in 1997. The score comes with a solo violin part, a *régie informatique* and a technical manual. In the latter, the designer of the first versions of the system, Andrew Gerzso, describes the system and how to build it 'from the score'.

At any moment in the piece, the sound from the violin is processed in a number of ways and/or extra sound samples are played back. All this is then projected in the space over a 2D sound projection system. This serves three purposes: extending the sound palette of the instrument, extending the compositional structures, and to introduce a spatial element into the composition [2, p.10].

[3] Published on Brilliant Classics, nr. 94842BR.

3.2 Processing Modules

The sound of the violin is processed by a number of digital signal processing (DSP) units. In every part of the piece a number of different combinations of units is used, resulting in the following modules:

- FS: frequency shifter;
- FSD: frequency shifter with delay;
- 6FS: six frequency shifters;
- 6FSD: six frequency shifters with delay;
- 2RMC: two ring modulators mixed to one comb filter;
- IR: 'infinite' reverberation;
- HR: harmoniser;
- 2HR: two harmonisers;
- 4HR: four harmonisers;
- HRD: harmoniser with delay;
- 4HRD: four harmonisers with delay;
- S: sampler;
- S-IR: sampler with infinite reverb.

The latter two do not transform the violin sound, but rather play back pre-recorded violin samples and sinusoids.

There is a maximum of six simultaneously playing modules that are routed to so-called 'sources' that go into the spatialisation system.

3.3 Sound Projection

3.3.1 Systems Used

According to the manual, the original version uses a six speaker set-up, plus extra speakers for the amplification of the violin sound. This six speaker setup is basically an equally distributed eight speaker surround setup with the front and rear speaker omitted. The latter two positions are also referred to in the score and moving sound will pass through these positions. The spatialisation system used in the original version [10] compensates for these gaps.

3.3.2 Movements

Besides the six main positions front-left (FL), middle left (ML), back-left (BL), back-right (BR), middle right (MR) and front-right (FR), the score prescribes a number of standardised movements:

- choose a random position from the main positions;
- choose a random position from a specified group of positions;
- choose a random position from the main positions every n milliseconds until further notice;
- go from B to F in a continuous movement, either via BL-ML-FL or BR-MR-FR;
- start a rotation in a random or specified direction of a specified length, starting from the current or a specified position;
- sweep back and forth between two positions in a specified time until further notice.

3.3.3 The Perception of Space

The technical manual describes an approach towards the perception of space in which the composition doesn't imply a specific speaker layout. It does so by introducing three main parameters, i.e. direction, presence in terms of distance and presence in terms of the perception of space. The previous may be true for direction, the latter two, however, are translated to more technical terms: direct sound, early reflections level, reverberation level and reverberation time, all of which are used as compositional parameters in the piece. So instead of composing with explicitly defined distance and space, leaving the implementation to the system designer, technical parameters are used compositionally, as with the processing system, and a specific approach towards reverberation is explicitly defined.

Imagine the use of for example Wave Field Synthesis (WFS) [11], as suggested in the technical manual [2, p.3]. In this system, the perceived distance is created in a completely different way, one could even say that it's the *raison d'être* of WFS (the description of which goes beyond the scope of this text). Distance, however, is not a parameter in the score, it results from the reverberation settings. Consequently, the algorithms in WFS responsible for distance will be hardly of use and the WFS-system will be using only a small part of its resources. Hence we feel that the spatialisation of this piece is about surround projection, compositionally but also to some extent technically, and that it isn't as much setup-independent as the technical manual suggests.

3.4 Cueing

In the score, every time a parameter change happens (meaning that processing module or spatialisation settings are changed or running dynamic processes are stopped), a cue is notated. Cues serve as a way of telling the system 'where we are'. This can be done manually (button, keystroke or mouse click, for example), however, for a number of parts of the composition score following [12,13] is necessary [2, p.5].

4. EXISTING VERSIONS

4.1 The Original Version

The very first version, that was used for the premiere at the Donaueschingen Festival in 1997, employs a NeXT computer with three ISPW processing boards running at a 32 kHz sample rate, running Max 0.26 and controlling an AKAI sampler. Already here, a version of Spatialisateur [10] was used.

Shortly after, a new version was built, running jMax on two Silicone Graphics Octane bi-processor computers, interconnected via MIDI. This version was used until 2004 and for recordings for Deutsche Grammophon.

In 2005 a third version saw light: two Apple G4's, connected via Ethernet for control syncing, running Max/MSP 4.5. More versions would follow and at least from 2008 on, the system ran on just one computer.

Since quite a long time, Ircam has been working on the development of score following systems, now known as ANTESCOFO [14], versions of which are used in the different original systems. System updates for Anthèmes 2 and the development of ANTESCOFO go hand in hand and (parts of) Anthèmes 2 are used for tests and demonstrations[4].

4.2 Our Work

In November 2007 we started working on our first version of the software. This was done in the context of our internship at the Conservatory of Amsterdam (CvA). In February and March 2008 we put on stage two performances, the last one at the Expert Meeting at the CvA, organised by Jos Zwaanenburg. Later that year, in December, we performed it twice more. After this the project ended, for other projects took over our attention. In 2015 we started working again, aiming to put on a series of performances, again with Marleen Wester. The plan is to investigate the current versions and performance practice, review our work from 2008 and finally start work on a new version, addressing issues such as performability, proprietary software, workflow and system design.

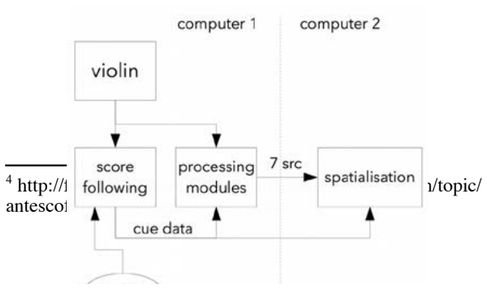
5. FIRST REMAKE

As mentioned above, our version is the first, and as of yet only, alternate version of the software for Anthèmes 2. In this section we will describe the several components of our system and the setup we used.

We would like to emphasize that we built the system as part of our internship as undergraduates. References to existing techniques in this paper are made in retrospect. In reality, everything was built with our best knowledge of that time, without paying much attention to existing work, other than the programming environment that we worked in.

5.1 Framework

The system is built in Max 5. See Figure 1 for an overview. In the process of building it, we found out that we needed to split the system over two computers. We decided to run the main system, including processing and score following, on the master computer, the second computer runs the spatialisation, listening to control messages from the master. This communication runs over an Ethernet connection. Both computers are equipped with a MOTU828 audio interface, interconnected with ADAT.



[4] http://l
antesco

Figure 1: System overview

A maximum of six sources sound simultaneously. We also made the amplification of the violin part of the system, so a total of seven sources are defined, meaning that seven channels of audio are sent to the spatialisation computer.

During performance, a human being, called the supervisor, keeps an eye on the system and interferes whenever necessary.

For the violin sound we used a DPA clip-on instrument microphone. We also introduced a Schertler contact microphone in the bridge of the violin for the score following, to minimise the influence of disturbing environmental sounds or sound from the speakers feeding back into the microphone.

In early versions we worked with a foot pedal used by the violin player to advance sections. This didn't work so well for the violin player, since it distracted her from her playing, so we decided to let the supervisor control this. To make sure there could be no misunderstanding whether or not the system was ready for the next section, we introduced a little LED at the bottom of the music stand which turns on when the system is readily waiting for the violin to start, and turns off consequently.

In terms of cueing we made a distinction between cue changes and section changes. On section change, the processing system, the spatialisation and the score following load new parameter data, corresponding to that section's cues. Also the processing modules' output signal routing changes (but stays static for that section). Cue changes are generated by the score following and sent to the processing system and the spatialisation and are also used by the score following system itself.

For rehearsal purposes we extended the system so one can skip to any cue at any time. This turned out to be of great importance to make the rehearsals a success. We would even go as far as to say that this should be a requirement for any such system.

5.2 Processing System

The processing modules mentioned in section 2.2 are implemented in separate patches. Using the Max poly~ object, processing modules that are not in use are turned off to save CPU-power. The outputs of the processing modules are routed to one of the seven sources mentioned, according to the schematics in the technical manual [2, p. 5-8].

5.3 Score Following

An overview of the score following system is shown in Figure 2.

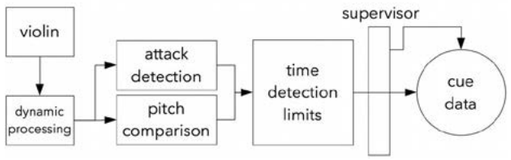


Figure 2: Score following overview

For the score following we use the Schertler input. This is preprocessed by running it through a spectral gate. Since we don't use this signal for any purposes other than score following, the reduction of audio quality due to this processing is no issue.

In order to achieve an accurate score following system, several methods of detection are stacked. The score following detects cues through the detection of volume or pitch changes.

Because of the preprocessing, attack detection can be very accurate. Some sections, however, have cues that are in the middle of a series of bowed notes, reducing effectiveness of attack detection. For these cues a pitch comparison algorithm is used. If a detected pitch falls within a predefined range, a cue is sent. In some cases, we needed to define a series of pitches to be detected, to reduce the chance of a detected cue before the related note is played. Pitch detection was implemented using the, now obsolete, fiddle~ Max external. Only specific sections make use of pitch detection, most cues can be accurately detected using just attack detection.

Furthermore, detection is only allowed within set timeframes. This is implemented using an opening and closing gate. The timing of this needed to be defined beforehand. To achieve this, the complete violin part was recorded, and all cue data was entered in an audio editor. A simple export of these cues resulted in the complete temporal data of all cues. These files were made relative, i.e. every cue time was counted not from the beginning of a section, but from the previous cue.

In practice, the speed with which sections are played changed during the rehearsal period. Therefore, timed detection gates weren't accurate anymore. This was compensated for by introducing a scaling factor for each cue. This is comparable to the virtual vs. real time mechanism, described in [13].

On top of these different detection layers, there is the manual layer, i.e. the control of the score following by the supervisor. Although the detection is programmed to be accurate, there is always the chance of a glitch or wrong choice of the computer program. During the performance, the supervisor is responsible for following the score and making sure the cues are sent correctly. This can be done by closing another gate to stop detected cues. This gate comes after all other gates and detection. Several shortcuts on the keyboard are programmed to either hold back any detected cues, or send a cue if a cue is missed, so everything is again synchronised.

It has never been the intention to use the score following without a supervisor. Although in some situations this system is operating very accurately, especially the attack detection, a supervisor knowing and following the score is necessary [15] to make definite choices.

5.4 Spatialisation

5.4.1 Nature of the System

The spatialisation system was built completely from scratch, because we felt that to challenge the intentions of the authors, we should not use existing algorithms. Also, we based our design on the instructions in the score, rather than on ideas of what comprises a complete, versatile spatialisation algorithm.

5.4.2 Speaker Setup and Amplification

We use a classic eight speaker setup, mainly because the eight speaker positions are explicitly used in the composition (including front and back) as directional parameters.

We also decided to not amplify the violin sound over separate speakers, but to mix this with the signal coming from the front-left and front-right speakers. In practice, we found that not every performance needed the amplification.

5.4.3 System Design

For each source a series of speaker output levels is calculated, based on the position parameter. A cosine envelope function is used for equal power distribution and a width factor has to be defined. The greater the distance between speakers, the bigger the width factor has to be to result in a correct distribution of the sound. Hence the necessity to be able to adjust this setting before performance, depending on the room in which the performance takes place and the way the speakers are set up.

The movement types are preprogrammed. Based on the list in section 2.3.2 a total of twelve movement types are defined, which are used to project the sources, according to cue information.

6. CONCLUSIONS

Reconstructing the software based on this score worked out really well in this case. We were able to compare our work with the Ircam version through attending a concert at the Louvre museum in Paris on November 21st, 2008. Although this cannot be called an objective observation in any way, we still think it's worth noting that in our opinion our version wasn't inferior at all. This success is of course partly due to the quality of the score and the clarity of the descriptions and instructions in the technical manual.

In terms of preventing live-electronic music computer programs from becoming obsolete, it's hardly possible to make generalisations about the validity of the described approach, simply because this paper only describes one successful attempt of one composition. This work can therefore not serve as proof that the approach suggested with the publication of the score of Anthèmes 2 is a feasible one per se. Nevertheless, we think it deserves more attention from the side of publishers and promoters, as well as composers and developers, to enable an informed appreciation of this approach.

7. FUTURE WORK

Our ambition is to make a new version of Anthèmes 2's software, using a truly open source musical programming language, to address compatibility issues, and perform with this in a series of concerts. It may also be an interest-

ing experience, because both existing versions (the original Ircam version and ours) are built in Max.

Acknowledgments

We would like to thank Marcel Wierckx for his continuous support and advice, the Utrecht School of Music and Technology and the Conservatory of Amsterdam, the kind people at Ircam for their feedback and encouragement, Sjoerd van der Sanden for joining the team in 2007/2008 and last but not least Marleen Wester.

Our thoughts go out to friends and family of Pierre Boulez, who passed away on January 5th, 2016.

8. REFERENCES

[1] P. Boulez, “Anthèmes 2”, *Universal Editions 31160*, 2005.

[2] A. Gerzso, “Anthèmes 2 – technical manual”, *Universal Editions 31160b*, 2005.

[3] M. Puckette, “The Deadly Embrace Between Music Software and its Users.”, *Keynote address at the EMS Network Conference*, Berlin, 2014.

[4] X. Pestova, M.T. Marshall and J. Sudol, “Analogue to digital: Authenticiy vs. sustainability in Stockhausen’s MANTRA (1970).”, P*roceedings of the International Computer Music Conference*, Belfast, 2008, pp. 201-204.

[5] C. Burns, “Realizing Lucier and Stockhausen: Case Studies in the Performance Practice of Electroacoustic Music.”, *Journal of New Music Research,* 31.1, 2002, pp. 59-68.

[6] A. de Sousa Dias, “Case studies in live electronic music preservation: Recasting Jorge Peixinho's Harmónicos (1967-1986) and Sax-Blue (1984-1992).”, *Journal of Science and Technology of the Arts,* 1.1, 2009, pp. 38-47.

[7] R. Esler, “Digital Autonomy in Electroacoustic Music Performance: Re-Forging Stockhausen.”, *Proceedings of the International Computer Music Conference*, New Orleans, 2006, pp. 131-134.

[8] M. Puckette, “New Public-Domain Realizations of Standard Pieces for Instruments and Live Electronics.”, *Proceedings of the International Computer Music Conference*, San Francisco, 2001, pp. 377-380.

[9] O. Baudouin, “A reconstruction of Stria.” *Computer Music Journal,* 31.3, 2007, pp. 75-81.

[10] J-M. Jot and O. Warusfel, “A Real-Time Spatial Sound Processor for Music and Virtual Reality Applications.”, *Proceedings of the 1995 International Computer Music Conference*, Banff, 1995, pp. 294-295.

[11] A.J. Berkhout, D. de Vries and P. Vogel, "Acoustic Control by Wave Field Synthesis", *The Journal of the Acoustical Society of America*, 93.5, pp. 2764-2778, 1993.

[12] B. Vercoe, “The Synthetic Performer in the Context of Live Performance”, *Proceedings of the 1984 International Computer Music Conference*, Barcelona, 1984, pp. 199-200.

[13] R.B. Dannenberg, “An On-line Algorithm for Real-Time Accompaniment”, *Proceedings of the 1984 International Computer Music Conference*, Barcelona, 1984, pp. 193-198.

[14] A. Cont, “ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music”, *Proceedings of the 2008 International Computer Music Conference*, Belfast, 2008, pp. 33-40.

[15] M. Puckette and C. Lippe, “Score Following in Practice”, *Proceedings of the 1992 International Computer Music Conference*, San Jose, 1992, pp. 698-701.

Stride: A Declarative and Reactive Language for Sound Synthesis and Beyond

Joseph Tilbian
jtilbian@mat.ucsb.edu

Andrés Cabrera
andres@mat.ucsb.edu

Media Arts and Technology Program
University of California, Santa Barbara

ABSTRACT

Stride is a declarative and reactive domain specific programming language for real-time sound synthesis, processing, and interaction design. Through hardware resource abstraction and separation of semantics from implementation, a wide range of computation devices can be targeted such as microcontrollers, system-on-chips, general purpose computers, and heterogeneous systems. With a novel and unique approach at handling sampling rates as well as clocking and computation domains, Stride prompts the generation of highly optimized target code. The design of the language facilitates incremental learning of its features and is characterized by intuitiveness, usability, and self-documentation. Users of Stride can write code once and deploy on any supported hardware.

1. INTRODUCTION

In the past two decades we have witnessed the rise of multiple open-source electronic platforms based on embedded systems. One of the key factors for their success has been in the simplifications made to programming their small computers.

By the nature of their design, they have mainly targeted physical computing and graphics applications. Audio has usually been made available through extensions. Although solutions leveraging existing operating systems and languages exist, what we have not seen yet is a full featured, audio-centric, multichannel platform capable of high resolution, low latency, and high bandwidth sound synthesis and processing. We attribute this to the lack of a high level domain specific programming language (DSL) targeting such a platform. All popular DSLs in the music domain have been designed to run on computers running full featured operating systems. Stride was conceived and designed to address this problem, enabling users to run optimized code on bare metal.

2. APPROACH

The field of DSLs for sound and music composition is old and crowded. To design a modern and effective language, multiple design requirements need to be addressed.

For the instrument designer, sound artist, or computer musician the language must simplify or unify the interface between language entities such as variables, functions, objects, methods, etc. It must simplify interaction programming and enable parallel expansion of its entities and interfaces.

From the perspective of digital signal processing, the language must be able to perform computations on a per sample basis, on real and complex numbers, in both time and frequency domains. It must also handle synchronous and asynchronous rates.

To take advantage of the current landscape of embedded and heterogeneous systems in an efficient manner, the language must abstract hardware resources and their configuration in a general and simple way. It must abstract the static and dynamic allocation of entities as well as threading, parallelism, and thread synchronization. It must also enable seamless interfacing of its entities running at different rates.

While designing Stride, the intuitiveness of the language as well as the experience of writing programs, by beginner and advanced users alike, topped the requirements mentioned above and both had a profound impact on its syntax design.

3. LANGUAGE FEATURES

A central consideration during the design of Stride was to treat the language as an interface and try to make it as “ergonomic” as possible. Two other criteria were *readability* and *flow*. That is, users should not need to read documentation to understand code and should be able to write code with as little friction as possible as the language works in a “physically intuitive” way similar to interfacing instruments, effects processors, amplifiers, and speakers in the physical world. To achieve this, features from popular and widely used general purpose and domain specific languages were incorporated into Stride, like:

- Multichannel expansion from SuperCollider [1]
- Single operator interface and multiple control rates from Chuck [2]
- Per sample processing and discarding control flow statements from Faust [3]
- Polychronous data-flow from synchronous and reactive programming languages like SIGNAL [4]
- Declarations and properties from Qt Meta Language
- Slicing notation for indexing from Python
- Stream operator from C++

The syntax of Stride is easy to learn as there are very few syntactic constructs and rules. Entities in the language are self-documenting through their properties, which expose the function of the arguments they accept. The choice of making Stride declarative was to separate semantics from any particular implementation.

The novel and unique aspect of Stride is making *rates* and *hardware computation cores* an intrinsic part of the language by introducing *computation domains* and synchronizing rates to them. This concept enables the distribution of various synchronous and asynchronous computations, encapsulated within a single function or method, to execute in different interrupt routines or threads on the hardware. The domains can potentially be part of a heterogeneous architecture. Rather than just being a unit generator and audio graph management tool, Stride enables the user to segment computations encapsulated in a unit generator during target code generation while handling it as a single unit in their code. Stride also features reactive programming, which enables complex interaction design.

This document presents a broad introduction to Stride, leaving many details out in the interest of space.

3.1 Language Constructs

There are two main constructs to the language: *Blocks* and *Stream Expressions*. Blocks are the building entities of the language while stream expressions represent its directed graph.

3.1.1 Blocks and Stream Expressions

Blocks are declared through a block declaration statement. They are assigned a *type* and a unique *label*. Labels must start with a capital letter and can include digits and the underscore character. A block's properties, discussed in detail in § 3.1.3, are part of the declaration and define its behavior. Code 1 shows a block declaration statement of type *signal* with default property values. The signal block is labeled *FrequencyValue*.

```
1 signal FrequencyValue {
2   default:  0.0
3   rate:     AudioRate
4   domain:   AudioDomain
5   reset:    none
6   meta:     none
7 }
```

Code 1: A *signal* block declaration statement with default properties

Blocks exchange *tokens* either synchronously or asynchronously through *ports*. Tokens represent a single numeric value, a Boolean value, or a character string. The number and types of ports of a block depends on its type. A block has primary and secondary ports. Primary ports are accessible through a block's label while secondary ports are accessible through its *properties*. Connections between primary ports are established in stream expressions using the *stream operator* (>>). Connections between primary and secondary ports are either established during a block's declaration or during invocation in stream expressions.

Code 2 is a stream expression where the primary ports of the *Input*, *Process*, and *Output* blocks are connected using the stream operator. A secondary port of the *Process* block, exposed through a property called *control*, is connected to a primary port of the *Value* block.

```
1 Input >> Process ( control: Value ) >> Output;
```

Code 2: A stream expression with four block connections

Stream expressions must end with a semicolon. They are evaluated at least once from left to right and in the top-down order in which they appear in the code.

3.1.2 Block Types

Block *types* are categorized into three groups: *Core*, *Auxiliary*, and *Modular*.

The core blocks are *signal*, *switch*, *constant*, *complex*, *trigger*, and *hybrid*. The signal block, discussed in detail in § 3.1.4, is the principal element of the language. It dictates when (the rate of token propagation) and where (the computation domain) computations occur within a stream expression. The *switch* block abstracts a toggle switch. It is asynchronous and can have one of two states: *on* or *off*, both keywords in Stride. The *trigger* block can trigger *reaction* blocks, allowing reactive programming within an otherwise declarative language. The *complex* block represents complex numbers and facilitates performing computations on them. The *hybrid* block enables the abstraction of port types allowing compile time type inference, akin to templates in object-oriented languages.

The auxiliary blocks are *dictionary* and *variable*. The dictionary block holds key and value pairs. The variable block dynamically changes the size of core block bundles, discussed in §3.1.5, enabling dynamic memory management.

The modular blocks are *module* and *reaction*. They encapsulate blocks and stream expressions to create higher level functions and reactions respectively. Unlike module blocks which operate on one token at a time, reaction blocks, when triggered, continuously execute until stopped when certain criteria are met.

3.1.3 Ports and Tokens

Ports have a direction and a type. A port's direction can either be *Input* or *Output*. Blocks receive or sample tokens through input ports and broadcast them through output ports. There are eight port types in total. A port's type is defined by two attributes. Each attribute is an element from the following two sets: {*Constant*, *Streaming*} and {*Real*, *Integer*, *Boolean*, *String*}.

The validity of connections between ports is determined by their types. Automatic type casting takes place between certain port types. Only a single connection can be established with an Input port while multiple connections can be established with an Output port. Constant Output ports can be connected to Streaming Input ports but Streaming Output ports can not be connected to Constant Input ports. Real, Integer and Boolean ports can be connected to each other but not to String ports and vice versa. Boolean Output port tokens are treated as 0 or 1 at Integer Input ports and as 0.0 or 1.0 at Real Input ports. Integer and Real Output port tokens with values 0 or 0.0 respectively are treated as false at Boolean Input ports while tokens with any other value are treated as true. Integer Output port tokens are cast to real at Real Input ports while Real Output port tokens are truncated at Integer Input ports.

3.1.4 The Signal Block

The signal block has five properties as shown in Code 1. The *default* property sets the block's default value as well as the primary port types to Streaming Integer, Streaming Real, or Streaming String depending on whether the value is an integer, a real or a string. The value assigned to the *rate* property sets the block to run either in synchronous or asynchronous mode. When assigned an integer or a real value it runs in synchronous mode and when assigned the keyword *none* it runs in asynchronous mode. The *domain* property sets the computation domain of the block and synchronizes its rate to the assigned domain's clock. The *reset* property resets the block to its default value when a trigger block assigned to it is triggered. All blocks in Stride have a *meta* property used for self documentation. It can be assigned any string value.

3.1.5 Block Bundles

Blocks can be bundled together to form *block bundles*. The primary ports of the bundled blocks are aggregated to form a single interface. Individual ports, or a set of ports, of the interface can be accessed by indexing. Indexing is not zero-based, but starts at 1. The square brackets are the bundle indexing and bundle forming operator. Core block bundles can be formed during declaration by specifying the bundle size in square brackets after the block's label. Bundles can also be formed in stream expressions by placing blocks or stream expressions in square brackets separated by commas.

3.2 Platforms and Hardware

Since Stride is a declarative language, a backend is required to translate Stride code to one that can be compiled and executed on hardware. A backend is known as a *Platform*. Stride code should start with the line of code shown in Code 3. It instructs the interpreter to load specific platform and hardware descriptor files. The platform descriptor file abstracts hardware resources and contains translation directives while the hardware descriptor file lists the available resources. When versions are not specified the latest descriptor files are loaded. A third file, the hardware configuration file, contains resource configurations. It can be specified after the hardware version in Code 3 using the keyword *with*. The default configuration file is loaded when nothing is specified. The descriptor and configuration files are written in Stride.

```
1 use PLATFORM version x.x on HARDWARE version x.x
```

Code 3: Loading a platform and a target hardware

The abstraction of hardware resources happens through blocks with reserved labels. These abstractions are common among all platforms. For example, *AudioIn* and *AudioOut* are signal bundles which abstract the analog and digital audio inputs and outputs of hardware. The constant block *AudioRate* abstracts the default sampling rate of these inputs and outputs, while the constant block *AudioDomain* abstracts the default audio callback function. *ControlIn*, *ControlOut*, *ControlRate*, and *ControlDomain* abstract non-audio ADCs, DACs, their default sampling rate, and related default callback function respectively. The range of *AudioIn* and *AudioOut* is [-1.0, 1.0], while that of *ControlIn* and *ControlOut* is [0.0, 1.0]. *DigitalIn* and *DigitalOut* are switch block bundles that abstract digital I/O TTL pins respectively. Communication protocols such as Serial, Open Sound Control [5], MIDI, etc. are also abstracted.

Creating aggregate systems based on multiple hardware platforms is also possible. This is achieved by abstracting the resources of aggregated hardware platforms through a single hardware descriptor file and by abstracting the communication between these platforms by the stream operator and the hardware configuration file.

3.3 Rates and Domains

A signal block is assigned a rate and a domain at declaration. Every domain has a clock with a preset rate derived from the hardware configuration file and abstracted through a constant block as discussed in § 3.2.

When a signal block is running in synchronous mode, it synchronizes itself with the clock of its assigned domain. It samples tokens at its primary input port and generates them at its assigned rate. In this mode, the signal block operates like a sample and hold circuit operating at a preset rate. When running in asynchronous mode, the signal block simply propagates tokens arriving at its primary input port. In both modes, all blocks (or stream expressions containing blocks) with connections to the signal block's primary output port recompute their state when a new token is generated. Computations happen in the domain each block is assigned to.

In Stride, rates and domains propagate through ports. The propagation is upstream. The keywords *streamRate* and *streamDomain* represent the values of the propagated rate and domain respectively where they appear. In Code 4 the values of *streamRate* and *streamDomain* in the *Map* module get their values from the *FrequencyValue* signal block.

```
1  ControlIn[1]
2  >> Map (
3    minimum: 55.0
4    maximum: 880.0
5  )
6  >> FrequencyValue;
7
8  Oscillator (
9    type:      'Sine'
10   frequency: FrequencyValue
11 )
12 >> AudioOut;
```

Code 4: A control input controlling the frequency of a sine oscillator

Since *FrequencyValue*, in Code 4, was not explicitly declared, it is treated as a signal block with default property values by the interpreter, as shown in Code 1. Therefore, in the *Map* module block the values of *streamRate* and *streamDomain* are *AudioRate* and *AudioDomain* respectively.

The *Oscillator* module in Code 4 encapsulates four signal blocks: *FreqValue*, *PhaseInc*, *Phase*, and *Output*. They represent the frequency, phase increment, phase, and output of the oscillator respectively. In the module's declaration, the rate of the first two signals is set to *none* and both are configured to receive their domain assignment from the block connected to the *frequency* property. The rate of the

Phase signal is set to *none* and is configured to receive its domain from the primary output port of the module, while the *Output* signal is configured to receive both its rate and domain from that port. This is summarized in Table 1.

Label	Rate	Domain
FreqValue	none	from 'frequency'
PhaseInc	none	from 'frequency'
Phase	none	streamDomain
Output	streamRate	streamDomain

Table 1: Labels, rates, and domains of signal blocks encapsulated in the *Oscillator* module

Unlike other DSLs, where unit generators represent a single computation unit, Stride can separate and distribute the constituent computations of its modules, such as *Oscillator*, to achieve extremely efficient and highly optimized target code.

To demonstrate the fine control Stride gives its user over generated code, consider a hypothetical platform which generates code [1] like the one shown in Code 5 based on Code 4. The hypothetical platform defines two domains: *AudioDomain* and *ControlDomain*. They are associated with the *audioTick* and *controlCallback* functions in the generated code respectively.

```
AtomicFloat ControlValue = 0.0;

void controlCallback (float *input, int size){
  ControlValue = input[0];
}

void audioTick (float &output){
  static float Phase, FreqValue, PhaseInc = 0.0;

  FreqValue = map(ControlValue, 55., 880.);
  PhaseInc = 2 * M_PI * FreqValue / AudioRate;

  output = sin(Phase);
  Phase += PhaseInc;
}
```

Code 5: Computations performed in the audio tick function on every call

The generated code is not efficient since *FreqValue* and *PhaseInc* are repeatedly computed for every audio sample. By explicitly declaring *FrequencyValue* as signal block and assigning it a slower rate, as shown in Code 6, the efficiency improves as shown in Code 7, where only changes to Code 5 are shown. This is equivalent to control rate processing in Csound and SuperCollider.

```
signal FrequencyValue { rate: 1024. }
```

Code 6: The rate of *FrequencyValue* set to 1024 Hz

```
Accumulator compute(1024. / AudioRate);

void audioTick (float &output){
  ...
  if (compute()){
    FreqValue = map(ControlValue, 55., 880.);
    PhaseInc = 2 * M_PI * FreqValue / AudioRate;
  }
  ...
}
```

Code 7: Accumulator added to reduce computation

The amount of computation can be further reduced by setting the *rate* of *FrequencyValue* to *none* and adding the *OnChange* module as shown in Code 8. Some of the computation will now happen asynchronously and in a reactive fashion. That is, only when the value of *ControlIn[1]* changes some of the computation will be performed as shown in Code 9.

```
signal FrequencyValue { rate: none }

ControlIn[1]
>> OnChange ()
>> Map ( minimum: 55. maximum: 880. )
>> FrequencyValue;
```

Code 8: Enabling asynchronous computation

```
void audioTick (float &output){
 ...
 static float PreviousValue = 0.0;
 ...
  if (ControlValue != PreviousValue){
    FreqValue = map(ControlValue, 55., 880.);
    PhaseInc = 2 * M_PI * FreqValue / AudioRate;
    PreviousValue = ControlValue;
  }
  ...
}
```

Code 9: Some computation performed only with value change

By changing the domain of *FrequencyValue*, shown in Code 10, the computations related to *FreqValue* and *PhaseInc* are performed in a reactive fashion in the control callback, as shown in Code 11. This change results in a highly efficient audio tick function.

```
signal FrequencyValue {
  rate:   none
  domain: ControlDomain
}
```

Code 10: Domain of *FrequencyValue* set to *ControlDomain*

```
AtomicFloat PhaseInc = 0.0;

void controlCallback (float *input, int size){
  static float FreqValue, PreviousValue = 0.0;

  if (input[0] != PreviousValue){
    FreqValue = map(input[0], 55., 880.);
    PhaseInc = 2 * M_PI * FreqValue/ AudioRate;
    PreviousValue = input[0];
  }
}

void audioTick (float &output){
  static float Phase = 0.0;

  output = sin(Phase);
  Phase += PhaseInc;
}
```

Code 11: Highly efficient audio tick function

Generating highly efficient subroutines is crucial to optimize performance on some embedded devices, particularly ones that support instruction caching and equipped with a tightly-coupled instruction memory.

[1] The C code shown in Code 5, 7, 9, and 11 is for demonstration purposes only. The code has not been generated by a backend implementation and is not complete.

3.4 Flow Control

Since control flow is not one of Stride's syntactical constructs, it can be realized in two ways. The first is through switching, achieved by bundling stream expressions followed by indexing the aggregate interface. The second, through triggering reaction blocks, which loop through the stream expressions they encapsulate until they are terminated.

4. CODE EXAMPLES

In the following subsections we present a few examples to demonstrate some of the features and capabilities of Stride.

4.1 Multichannel Processing

In Code 12 the levels of the first two signal blocks of the *Input* block bundle are changed by two signal blocks *A* and *B*. They are then mixed down to a single signal connected to the input ports of the first two signal blocks of the *Output* block bundle, as depicted in Figure 1. All signal blocks are declared with default values.

```
signal Input [4] {}
signal Output [4] {}
signal A {}
signal B {}

Input[1:2]
>> Level ( gain: [ A, B ] )
>> Mix ()
>> Output[1:2];
```

Code 12: Selective multichannel level adjustment and signal mixing

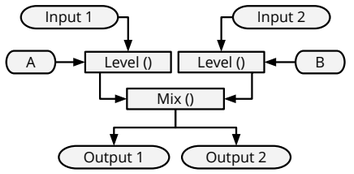


Figure 1: Selective multichannel processing

4.2 Generators, Envelopes, Controls, and a Sequencer

In Code 13 two sine oscillator module blocks, one oscillating at a perfect fifth of the other, are connected to two envelope generator module blocks. The *reset* ports of the oscillators and envelope generators are connected to the trigger block *Trigger*. This is depicted in Figure 2. When triggered, the oscillators' phase gets reset to zero (the default value) while the envelope generators restart. *Trigger* is activated on the rising edge of *DigitalIn[1]*.

```
constant Frequency { value: 440. }
trigger Trigger {}

DigitalIn[1] >> Trigger ;

Oscillator (
  type:       'Sine'
  frequency:  [ 1.0, 1.5] * Frequency
  amplitude:  [ 0.66, 0.33]
  reset:      Trigger
)
>> AD (
  attackTime: [ 0.6 , 0.8 ]
  decayTime:  [ 1.4 , 1.2 ]
  reset:      Trigger
)
>> Mix ()
>> AudioOut[1:2];
```

Code 13: Two sine oscillators connected to two attack / decay modules

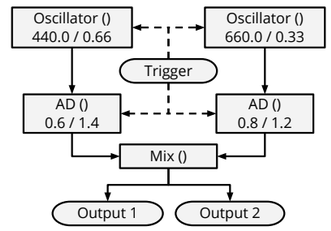


Figure 2: Sine oscillators and attack / decay modules with reset control

Code 14 extends Code 13 after modifying the block type of *Frequency*. The extension enables the control of the oscillators' frequencies through *ControlIn[1]*. When the value of *ControlIn[1]* changes, it gets mapped exponentially and smoothed at a rate 20 times less than the *AudioRate*.

```
signal Frequency { rate: AudioRate / 20. }

ControlIn[1]
>> OnChange()
>> Map ( mode: 'Exponential' minimum: 110.
maximum: 880. )
>> Smooth ( factor: 0.05 )
>> Frequency;
```

Code 14: Controlling the frequencies of the oscillators

Code 13 can also be extended by Code 15 after changing the block type of *Frequency* and reconnecting *Trigger*. The *ImpulseTrain* module block generates a trigger that triggers the *Sequencer* reaction block, whose values are imported from a Stride file called *Notes* into the *note* namespace. The file contains constant block declarations of musical notes.

```
import Notes as note

signal Frequency { default: note.C4 rate: none }

ImpulseTrain ( frequency: 0.5 )
>> ImpulseTrainValue
>> Compare ( value: 0 operator: 'Greater' )
>> Trigger
>> Sequencer (
  values: [ note.C4, note.E4, note.G4, note.C5 ]
  size:   4
  mode:   'Random'
)
>> Frequency;
```

Code 15: Control and triggering through an impulse train and a sequencer

4.3 Feedback

Code 16 is a feedback loop with 32 samples fixed delay as depicted in Figure 3. *Input* and *Feedback* signal blocks are

bundled together before being connected to *Level* module blocks. The mixed output is then delayed by 32 samples and streamed into *Feedback*.

```
[ Input, Feedback ]
>> Level ( gain: [ 0.50, -0.45 ] )
>> Mix ()
>> Output
>> FixedDelay ( samples: 32 )
>> Feedback;
```

Code 16: Feedback with 32 samples delay

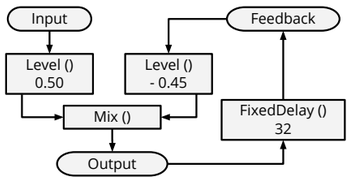


Figure 3: Feedback with 32 samples delay

4.4 Frequency Modulation Synthesis

Code 17 is a single oscillator feedback FM. The output of the oscillator controls its own frequency after being multiplied by a modulation index and offset by a base frequency. The index, base frequency, and amplitude are controlled by control inputs

```
signal Index     { rate: none }
signal Frequency { rate: none }
signal Amplitude { rate: none }

ControlIn[1:3]
>> OnChange ()
>> Map (
  mode:    [ 'Linear', 'Exponential', 'Linear' ]
  minimum: [ 0.08, 40.0 , 0.0 ]
  maximum: [ 2.00, 200.0, 1.0 ]
>> [ Index, Frequency, Amplitude ];

Oscillator (
  type:      'Sine'
  frequency:  Index * Output + Frequency
  amplitude:  Amplitude
)
>> Output;
```

Code 17: Single Oscillator Feedback Frequency Modulation

4.5 Fast Fourier Transform

Code 18 is a smoothed pitch tracker driving a sinusoidal oscillator. FFT is performed on a bundle and the magnitude of the spectrum is computed, followed by finding the index of the first maximum and converting it to a frequency value. These computations are performed at *AudioRate / Size* set by *PeakFrequency*. The computed frequency value is then smoothed at a faster rate to control the frequency of the oscillator running at *AudioRate*. *streamRate* represents the value of the rate of the output port of the *Level* module block, which is *AudioRate / Size*.

```
constant Size { value: 1024. }
signal InputBundle [Size] { rate: none }
signal PeakFrequency { rate: AudioRate / Size }
signal SmoothFrequency { rate: AudioRate / 20. }

InputBundle
>> RealFFT ()
>> ComplexMagnitude ()
>> FindPort ( at: 'Maximum' mode: 'First' )
>> Level ( gain:  streamRate / 2 )
>> PeakFrequency
>> Smooth ( factor: 0.01 )
>> SmoothFrequency;

AudioIn[1]
>> FillBundle ( size: Size )
>> InputBundle;

Oscillator ( frequency: SmoothFrequency )
>> AudioOut[1:2];
```

Code 18: FFT Peak Tracking

4.6 Multirate Signal Processing

In Code 19 a baseband signal with 8 KHz bandwidth sampled at 48 KHz is decimated by a factor of 4 before further processing is performed on it to reduce the number of computations. The signal is then interpolated back to the original sampling rate. The *DSP* module block is a placeholder for a chain of signal processing module blocks.

```
signal Input           { rate: 48000 }
signal ProcessedSignal { rate: 12000 }
signal Output          { rate: 48000 }

Input
>> Decimation (
  type:        'PolyphaseFIR'
  baseband:    8000
  attenuation: 60
  factor:      4
)
>> DSP ()
>> ProcessedSignal
>> Interpolation (
  type:        'PolyphaseFIR'
  bandwidth:   8000
  attenuation: 60
  factor:      4
)
>> Output;
```

Code 19: Multirate processing by decimation and interpolation

4.7 Granular Synthesis

In Code 20 grains are formed using sine oscillators and Gaussian envelopes. The oscillators and their corresponding envelopes are triggered by the *GrainState* switch block bundle through the *SetPort* module block, which acts as a demultiplexer. The *index* of the *SetPort* module is controlled by the *Counter* module block, which increments at the *GrainTriggerRate* value and rolls over when it reaches the *NumberOfGrains* value. The state of a grain is reset after its envelope has generated the required number of samples, computed from the *GrainDuration* value. The output of the envelopes are then mixed and sent to the audio output after adjusting the level.

```
constant NumberOfGrains   { value: 50 }
constant GrainTriggerRate { value: 15 }
constant GrainDuration    { value: 0.005 }
constant GrainFrequency   { value: 220 }

signal GrainIndex {
  default: 0
  rate:    GrainTriggerRate
}

trigger ResetGrainState [NumberOfGrains] {}

switch GrainState [NumberOfGrains] {
  default: off
  reset:   ResetGrainState
}

Counter (
  startValue: 1
  rollValue:  NumberOfGrains
  increment:  1
)
>> GrainIndex;

on
>> SetPort (
  index: GrainIndex
)
>> GrainState;

Oscillator (
  type:      'Sine'
  frequency: GrainFrequency
  reset:     GrainState
)
>> Envelope (
  type:     'Gaussian'
  size:     GrainDuration * streamRate
  start:    GrainState
  complete: ResetGrainState
)
>> Mix ()
>> Level ( gain: 1.0 / NumberOfGrains )
>> AudioOut[1:2];
```

Code 20: Synchronous triggering of statically allocated grains

Advanced granular synthesizers can be designed in Stride by allocating grains dynamically using the *variable* block to manage the size of core block bundles and triggering them using *reaction* blocks.

5. CONCLUSIONS

With its many features, Stride is an ideal language for creating and deploying new musical instruments on embedded electronic platforms. With few syntactic constructs, it is easy to learn, while its readability and intuitive coding flow make it an attractive choice for beginners and experienced users alike.

Stride documentation is available at:

`http://docs.stride.audio`

Acknowledgments

This work was funded in part by a graduate fellowship by the Robert W. Deutsch Foundation through the AlloSphere Research Group.

6. REFERENCES

[1] J. McCartney, "SuperCollider: a new real time synthesis language," in *Proceedings of the 1996 International Computer Music Conference*, Hong Kong, 1996.

[2] G. Wang and P. R. Cook, "ChucK: A Concurrent, On-the-fly, Audio Programming Language," in *Proceedings of the 2003 International Computer Music Conference*, Singapore, 2003.

[3] Y. Orlarey, D. Fober, and S. Letz, "Syntactical and semantical aspects of Faust," *Soft Computing*, vol. 8, no. 9, pp. 623–632, 2004.

[4] A. Gamatié, *Designing Embedded Systems with the SIGNAL Programming Language*. Springer, 2010.

[5] M. Wright and A. Freed, "Open Sound Control: A New Protocol for Communicating with Sound Synthesizers," in *Proceedings of the 1997 International Computer Music Conference*, Thessaloniki, 1997.

Embedding native audio-processing in a score following system with quasi sample accuracy

Pierre Donat-Bouillud
IRCAM UMR CNRS STMS 9912,
INRIA Paris - MuTant Team-Project
ENS Rennes
pierre.donat-bouillud@ens-rennes.fr

Jean-Louis Giavitto
IRCAM UMR CNRS STMS 9912,
INRIA Paris - MuTant Team-Project
jean-louis.giavitto@ircam.fr

Arshia Cont
IRCAM UMR CNRS STMS 9912,
INRIA Paris - MuTant Team-Project
arshia.cont@ircam.fr

Nicolas Schmidt
Computer Science Dept.,
Pontificia Universidad Catolica de Chile
nschmid1@uc.cl

Yann Orlarey
Grame, Lyon, France
orlarey@grame.fr

ABSTRACT

This paper reports on the experimental native embedding of audio processing into the Antescofo *system, to leverage timing precision both at the program and system level, to accommodate time-driven (audio processing) and event-driven (control) computations, and to preserve system behaviour on multiple hardware platforms. Here native embedding means that audio computations can be specified using dedicated DSLs* (e.g., *Faust) compiled on-the-fly and driven by the* Antescofo *scheduler. We showcase results through an example of an interactive piece by composer Pierre Boulez,* Anthèmes 2 *for violin and live electronics.*

1. IMS AND EVENT-DRIVEN *VS.* TIME-DRIVEN ARCHITECTURES

Interactive Music Systems (IMS) were promoted in early 1990s in an attempt to enable interaction between human musicians and real-time sound and music computing, initially for applications in *mixed music* defined by association during live music performance of human musicians and computers [15].

One specific challenge of IMS is to manage two "*time domains*": asynchronous event-driven computations and time-driven periodic management of audio processing. It led to the development of real-time graphical programming environments for multimedia such as *Max* [13] and the open source *PureData* [12].

In event-driven systems, processing activities are initiated as a consequence of the occurrence of a significant event. In time-driven systems, activities are initiated periodically at predetermined points in real-time and lasts. Subsuming the event-driven and the time-driven architectures is usually achieved by embedding the event-driven view in the time-driven approach: the handling of control events is delayed and taken into account periodically, leading to several internally maintained rates, *e.g.*, an audio rate for audio, a control rate for messages, a refresh rate for the user-interface, *etc.* This approach is efficient but the time accuracy is *a priory* bounded by the control rate.

An example of this approach is Faust [10], where control events are managed at buffer boundaries, *i.e.* at the audio rate. In Max or *PureData*, a distinct control rate is defined. This control rate is typically about 1ms, which can be finer that a typical audio rate (a buffer of 256 samples at sampling rate of 44100Hz gives an audio rate of 5.8 ms), but control computations can sometimes be interrupted to avoid delaying audio processing (*e.g.* in Max). On the other hand, control processing can be performed immediately if there is no pending audio processing.

The alternative is to subsume the two views by embedding the time-driven computations in an event-driven architecture. As a matter of fact, a periodic activity can be driven by the events of a periodic clock. [1] This approach has been investigated in ChucK [17] where the handling of audio is done at the audio sample level. Computing the next sample is an event interleaved with the other events. It results in a tightly interleaved control over audio computation, allowing the programmer to handle on the same foot signal processing and higher level musical and interactive control. It achieves a time accuracy of one sample. But this approach sacrifices the performance benefits of block-based processing (compiler optimizations, instruction pipelining, memory compaction, better cache reuse, *etc.*).

In this paper we propose a new architecture for the native embedding of audio computation in *Antescofo* [2]. The *Antescofo* system is a programmable score following system. *Antescofo* offers a tight coupling of a real-time machine listening [3] with a real-time synchronous Domain Specific Language (DSL) [7]. The language and its runtime system are responsible for timely delivery of *message passing* to host environments (MAX or *PureData*) as a result of reactions to real-time machine listening. The work presented here extends the *Antescofo* DSL with native audio processing capabilities.

[1] From this point of view, the only difference between waiting the expiration of a period and waiting the occurrence of a logical event is that, in the former case, a time of arrival can be anticipated.

In our approach, we do not attempt to provide yet-another universal language for audio processing but to provide the ability to compose complex architectures, using embedded local codes that employ specialized DSL, on-the-fly compilation and dynamic type-checking for passing between various time-domains. This approach reflects several important considerations: (1) to harness existing and well established audio processing systems, (2) to fill the gap between authorship and real-time performance; and (3) to improve both performance and time accuracy compared to existing IMS.

We start the paper by providing the necessary background on the *Antescofo* approach by focusing on a real-world example, an interactive piece by composer Pierre Boulez, *Anthèmes 2* for violin and live electronics, cf. Fig. 1. Section 3 discusses the main contribution of the paper by providing a time-aware semantics for combining real-time control and signal processing in *Antescofo*. Finally, we showcase results through the example of embedding audio processing in the Antescofo score of *Anthèmes 2*.

2. REAL-TIME COORDINATION OF MUSICAL EVENTS AND COMPUTING ACTIONS

2.1 A Paradigmatic Example

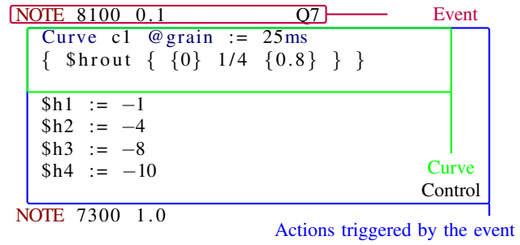
As an illustration, we showcase a mixed music piece that has entered the repertoire, *Anthèmes 2* (1997) by Pierre Boulez for violin and live electronics. The dominating platforms for programming such interactive paradigms are the graphical programming languages *Max* or its open-source counterpart *PureData*. In this section, we focus on a *PureData* implementation taken from *Antescofo*'s tutorial [5].

Programming of Interactive Music pieces starts by a specification of interactions, computing processes and their relations between each other and with the physical world in form of an *Augmented Music Score*. Fig. 1 (left) shows the beginning few bars of *"Anthèmes 2"*, Section 1. The top staff, upper line, is the violin section for human performer and in traditional western musical notation; and lower staves correspond to computer processes either for real-time processing of live violin sound (four *harmonizers* and *frequency shifter*), sound synthesis (two *samplers*), and spatial acoustics (artificial reverberation *IR*, and live spatialisation of violin or effect sounds around the audience). Computer actions in Fig. 1 are ordered and hung either upon a previous action with a delay or onto an event from a human performer. Computer Processes can also be chained (one sampler's output going into a reverb for example) and their activation is *dynamic* and depends on the human performer's interpretation.

Fig. 1 (right) shows the main patcher window implementation of the electronic processes of the augmented score in *PureData (Pd)*. The patch contains high-level processing modules, *Harmonizers*, *Samplers*, *Reverb*, *Frequency Shifting* and *Spatial Panning*, as sub-patchers. The temporal orderings of the audio processes is implicitly specified by a data-driven evaluation strategy of the data-flow graph. For example, the real-time scheduling mechanism in *PureData* system is mostly based on a combination of control and signal processing in a round-robin fashion [14], where, during a scheduling tick, time-stamped actions, then DSP tasks, MIDI events and GUI events are executed, cf. Fig. 2. Scheduling in *PureData* is thus block-synchronous, meaning that controls occur at boundaries of audio processing. Furthermore, in data-flow oriented language, the audio processes activation, their control and most importantly their interaction with respect to the physical world (human violinist) cannot be specified nor controlled at the program level.

2.2 Authorship in *Antescofo*

Real-time coordination and synchronization between human performer's events and computer actions is the job of *Antescofo* [2]. Code 1 shows the *Antescofo* excerpt corresponding to the augmented score in Fig. 1.



Code 1. Antescofo score for the first notes of *Anthème 2*

In the *Antescofo* code fragment above, notice the specification of both expected musical events from the physical environment (the NOTE keyword), computing actions (the sampling of a curve every 25ms for the next 1/4 beat) and their temporal relationships, *i.e.*, their *synchronization* (the sampling starts with the onset of the NOTE).

The *Antescofo* language provides common temporal semantics that allow designers and composers to arrange actions in multiple-time frameworks (absolute time, relative time to a tempo, event or time triggered), with multiple synchronization and error-handling strategies [4, 7]. Actions can be triggered simultaneously to an event detection by machine listening $e(t)$, or scheduled relatively to the detected musician's tempo or speed $\dot{e}(t)$. Actions can live in nested and parallel blocks or group, and the user can decide to program synchrony of a block on static or dynamic *target*s in the future (for instance, at the end of a phrase). Block contents can also be "continuous", as for the curve construct that performs a sequence of actions for each sample of a breakpoint function.

Real-time control in *Antescofo* follows a reactive model of computation assuming the *synchrony hypothesis* [1, 8]: atomic actions hooked directly to an event, called *reactions*, should occur in zero-time and in the right order. This hypothesis is unrealistic, however in practice, the system needs only to be quick enough to preserve the auditory perception of simultaneity, which is on the order of 20 milliseconds. This hypothesis is also adopted by ChucK [17] and makes the language a *strongly timed computer music language*:

- time is a first class entity included in the domain of discourse of the language, not a side-effect of the computations performance [9, 16];

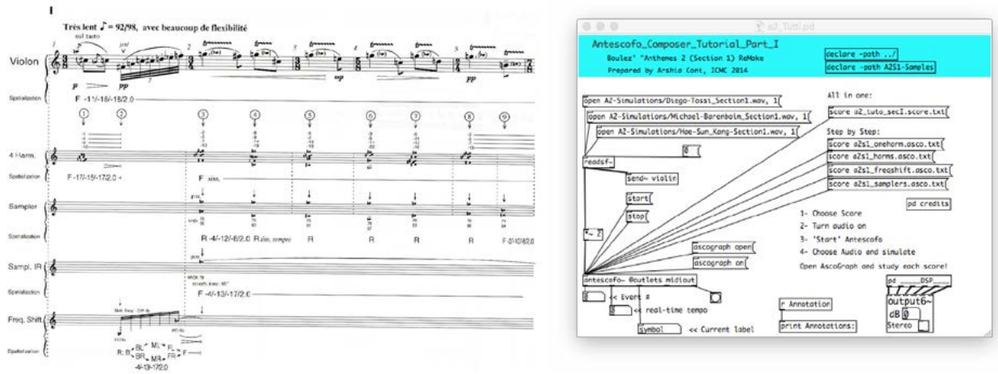


Figure 1. *Left:* Composer's score excerpt of *Anthèmes 2* (Section 1) for Violin and Live Electronics (1997). *Right:* Main *PureData* patcher for *Anthèmes 2* (Section 1) from *Antescofo Composer Tutorial*.

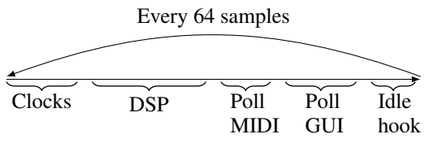


Figure 2. Scheduling cycle in *PureData* (polling scheduler)

- *when* a computation occurs is explicit and formalized in language semantics, ensuring behavior predictability and temporal determinism;
- assuming enough resources, the temporal behavior of a program is free from underlying hardware constraints and non-deterministic scheduling.

Antescofo synchronous programs departs in several ways from synchronous languages: *Antescofo* execution model manages explicitly the notion of duration, so the programmer may for instance trigger a computation after a well-defined delay. It is possible to refer to various time coordinates, including user-defined ones. And the dedicated language offers several constructs corresponding to "continuous actions" that span over an interval of time. Furthermore, *Antescofo* is dynamic and allows the dynamic creation of parallel processes.

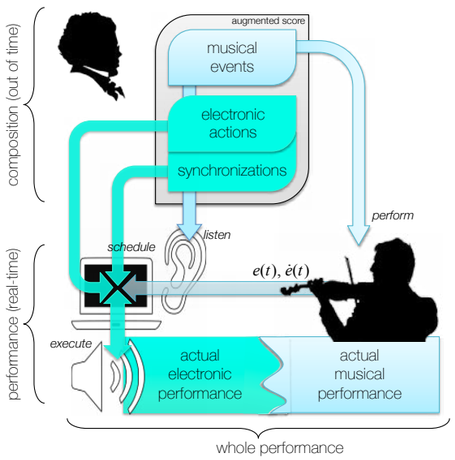


Figure 3. Antescofo Executation Diagram

The role of the *Antescofo* runtime system is to coordinate computing actions with physical events implementing the specified synchronizations, as shown in Fig. 3.

3. TIME-AWARE COMBINATION OF SIGNAL PROCESSING AND REAL-TIME CONTROL IN *ANTESCOFO*

During run-time execution, the standard *Antescofo* implementation delegates the actual audio computations to the host environment. So, their time-safety and consistencies are subject to real-time scheduling of control, signal processing, and other user interrupts such as external controls of the GUI in the host. *PureData* and Max's capability of combining real-time control and signal processing within the same framework is the major feature of their architecture for end-users, but presents several shortcomings. Time is implicit in the data-flow style, so some temporal constraints between audio and control computation are simply not expressible. And the round-robin scheduling strategy forces a fixed interleaving of control and audio computation, that reduces the temporal accuracy that can be achieved.

Embedding digital audio processing in *Antescofo* is an experimental extension of the language, aimed at driving various signal processing capabilities directly within *Antescofo*, to overcome these drawbacks. The rest of this section presents this extension, sketches the underlying execution model and discusses the resulting temporal accuracy.

3.1 Audio Processing Nodes and Their Types

Signal processors are defined directly in an *Antescofo* program, harnessing several signal processing libraries. Currently, *Faust* [10] and a few dedicated DSP processors can be defined. These examples are enough to validate the approach and its versatility.

Faust processors are defined directly as Faust programs within the *Antescofo* score. They are compiled by the embedded *Faust* compiler when the *Antescofo* score is loaded and the resulting C code is compiled on-the-fly (with the in-core LLVM compiler) into a dynamically linked function that implements the specified computation. A few dedicated DSP processors have been specifically developed, notably a FFT transformation based on the Takuya Ooura's FFT package. The objective is to validate the integration of spectral computations in the *Antescofo* audio chains, an example of time-heterogeneous audio computations.

For efficiency reasons, audio samples are grouped into buffers and all samples of a buffer are handled altogether by a DSP node which therefore performs its computation periodically: a buffer corresponds to a set of values that are supposed to be produced and consumed sequentially in time, but that are all accessible at the same moment because the actual use of the buffer is deferred to a later moment. So, irrespectively of the exact computation it achieves, a DSP processor can be abstracted by a function that processes a *sequence of buffers*. These sequences are characterized by a *buffer type* corresponding to the periodicity and the size of a buffer in the sequence. It represents also the information needed to associates a time-stamp to each element in the sequence once a time-stamp is given to the buffer. Such types make it possible to represent overlapping buffers, which are common when doing spectral processing.

Antescofo distinguishes between two kinds of DSP nodes, as illustrated on Fig. 4. *Isochronous* node, or *effects*, transform a buffer into a buffer of same type, so only the elements of the buffers are modified, but not the size nor its periodicity. They can have ordinary *Antescofo* variables as additional input or output control. Typically, Faust processors are isochronous.

Heterochronous nodes consumes and produces sequences of buffers of different types. A *detector* which takes an audio signal as input and outputs a boolean as the value of a control variable when some condition is satisfied, is an example of heterochronous node. A Fourier transformation is another example of a heterochronous computation.

3.2 Connecting Audio Processing Nodes

DSP nodes are connected by *links*. Links are implicit *buffer type adapters*: they are in charge of converting a sequence of buffers into another equivalent sequence of buffers, leaving the samples untouched. Equivalence means here that we have the same sequence of buffer elements, irrespectively of the buffer boundaries, or that the output sequence is an "(un)stuttering" of the input sequence in case of overlapping buffers. Links represent also input or output channels, that transport the audio signal from and to the soundcard or the host environment. Once the buffer types of DSP nodes are known, the adequate link adaptation can be automatically generated to convert between buffer types.

Links appears as a special kind of variable in *Antescofo*. They are denoted by $$-identifiers where ordinary variables are denoted by $-identifiers. As for ordinary variable, the occurrence of a link in a control expression denotes the "current value" of the corresponding buffer sequence, *i.e.* the sample corresponding to the current instant.

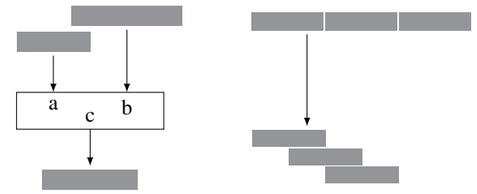


Figure 4. *(Left):* An isochronous node processes buffers, not sequences of buffers. This one has two input ports a and b and an output port c. *(Right):* A link act as an implicit type converter and transforms sequences of buffers into an equivalent sequence of buffers. Here, the input buffers are contiguous and the output buffers overlap.

3.3 Dynamic Patches

DSP nodes and links are declared independently in the score, and can be connected later using a new dedicated *Antescofo* action, *patch*, which represent a set of equations. One equation corresponds with one DSP node. Occurrences of links in the equations materialize the producer/-consumer relationships between DSP nodes. Fig. 5 shows the effects and the links for the DSP graph of the beginning of *Anthèmes 2*.

Patch actions can be launched dynamically, enabling reconfiguration of the audio computation graph in response to the events detected by the listening machine. These dynamic changes can be also synchronized with the musical environment using the expressive repertoire of synchronization strategies available in *Antescofo*.

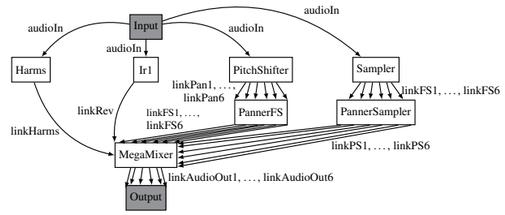


Figure 5. DSP graph at the beginning of *Anthèmes 2* by Pierre Boulez. The audio signal flows from *Input* to *Output*.

If a DSP node or a link channel is not used in an active *patch*, the link and the related DSP nodes are disabled as shown on Fig. 6: removing a link (resp. a node) from the audio graph also removes the subtree rooted by the link (resp. the node). All links and nodes that are not connected to an output channel are also disabled.

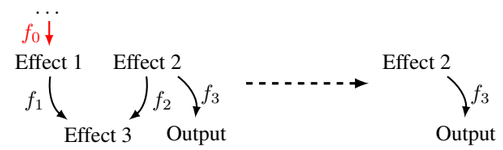


Figure 6. Removing the link f_0 in the DSP graph. As Effect 3 and Effect 3 need buffers traversing f_0, Effect 1, Effect 3 and link f_1 are removed from the graph. The incoming effects to Effect 1 don't have any other outcoming path to the Output, so they are also removed from the Dsp graph.

3.4 Architecture Rationals

Several benefits are expected with this tight integration of audio computations in the language: i) An augmented score will be specified by one textual file which records the definitions and the control of all the software components involved in the implementation of the piece. ii) The network of signal processors is heterogeneous, mixing DSP nodes specified with different tools (Faust , FluidSynth, etc.). iii) The network of signal processors can change dynamically in time following the result of a computation. This approach answers the shortcomings of fixed (static) data-flow models of the Max or Pd host environments. iv) Signal processing and its scheduling are controlled at a symbolic level and can be guided, e.g. by information available in the augmented score (like position, expected tempo, etc.). v) This tight integration allows concise and effective specification of finer control for signal processing, at a lower computational cost. One example (developed below) is the use of symbolic curve specifications to specify variations of control parameters at sample rate. vi) Signal processing can be done more efficiently. For example, in the case of a Faust processor, the corresponding computation is highly optimized by the Faust on-the-fly compiler.

3.5 A GALS Execution Model

The temporal interactions between audio and control computations within Antescofo can be roughly conceived as two autonomous worlds that evolve in parallel and that interact by shared information. Audio computation can be seen as computation on continuous data and control computation as sporadic processes. In a rough approximation, audio processing is done continuously and in parallel to the control computations.

To fully understand the interplay between audio and control computation, one has to refine the "continuous-processing-of-audio-signal" notion into the more realistic "sample-processing-of-sampled-audio-signal" implementation. At the end of the day, each sample corresponds to a physical date and moving forward in the buffer corresponds to some time progression.

A control signal (an external event, the exhaustion of a delay, etc.) may occur during an audio computation: control computations and audio computation are "asynchronous". But all audio computations correspond to well known time-stamps and control computations are also well-ordered by causality. Thus, locally, the computation appears "synchronous". The term GALS for "globally asynchronous, locally synchronous" has been used to describe this situation [6]. The challenge thus lies at the interface of the two models of computation : control computation which is supposed to happens instantly, may be delayed until the end of an audio computation, which decreases the temporal accuracy of the system.

3.6 Temporal Accuracy in a GALS Execution Model

If audio processing and control processing appear as parallel computations, they interact by sharing information or spanning computations at some points in time. In an idealistic world with unbounded computational power, buffer sizes will be reduced to only one sample and control and audio processing will be fully interleaved. This execution model achieves sample-accuracy , the greatest possible temporal accuracy between the two asynchronous audio and control worlds. Due to the limited computational power available, buffer sizes cannot be shrunken to one sample. To take advantage of the buffer processing approach, the Antescofo execution model takes into account the control variables during DSP processing only at some limited dates, as follows (cf. Fig. 7).

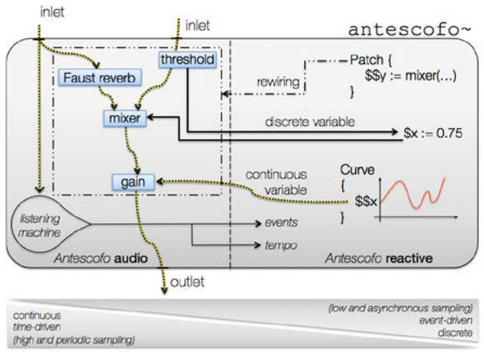


Figure 7 . Interaction between audio processing and reactive computations.

i) The dependencies between audio computations are ordered using a topological sort to sequence the buffer computations. If the audio computations are independent of any control parameter, this achieves sample-accuracy while preserving block computation. ii) Musical events are detected by the listening machine in an audio stream and signaled to the reactive engine which eventually triggers control computation at the end of the processing of the input audio buffer. The theoretical constraints of the spectral analysis done to detect the musical events imply that at most one event is recognized within roughly 11 ms (512 samples at 44 100Hz sampling rate). iii) Reactive computations that are not spanned by the recognition of a musical event, are triggered by the elapsing of a delay or by an external event signaled by the environment (e.g., a keyboard event). In these case, the temporal accuracy of the reaction is those provided by the host environment (typically in Max, 1 ms for an external event, usually better for the exhaustion of a delay). We say that these computations are system-accurate. iv) Computations in the reactive engine may start, stop or reorganize the audio computations. These actions take place always at the end of a reaction and at "buffer boundaries", that is, between two buffer processing in the DSP network. We say that these computations are buffer-accurate. v) The audio computation is controlled by discrete data or symbolic continuous data computed in the reactive engine. Discrete data are read by audio computation at buffer boundaries and are assumed constant during the next buffer computations while symbolic continuous data are handled as if they are read for the processing of each sample because their time-evolution is know a priori .

This approach is temporally more accurate than the usual approach where control processing is done after the processing of all audio buffers, for two reasons.

Control computation is interleaved with DSP node processing and is not delayed until the end of all DSP processing. Compared to sample-accuracy, it means that taking into account the change of a control variable is delayed for each audio node only by the time interval corresponding to its own rate. Because the DSP network includes heterogeneous rates, the benefit can be sensible. Furthermore, if the code corresponding to the DSP nodes permits (this is usually the case for *Faust* specified nodes), these rates can be dynamically adjusted to achieve a greater temporal accuracy (at the expense of more overhead) or to lower the computational cost (at the expense of temporal accuracy).

The second benefit of our approach is that control variables managed within the reactive engine can be taken into account during audio-processing at the level of sample-accuracy, when they are tagged "continuous" (this is the case when their identifier starts with $$). Continuous variable can be used as ordinary *Antescofo* control variables. However, when their updates can be anticipated, because for instance they are used to sample a symbolic *curve* construct, this knowledge is used to achieve sample accuracy in the corresponding audio processing. Fig. 8 illustrate the difference; the top plots draw the values of the variable $y in relative and absolute time in the program:

```
Curve @grain 0.2s { $y { {0} 6 {6} } }
```

This curve construct specifies a linear ramp in time relative to the musician tempo. For the implementation, the control variable $y samples the curve every 0.2 s (notice that the sampling rate is here specified in absolute time) going from 0 to 6 in 6 beats. There is 3 changes in the tempo during the scan of the curve, which can be seen as slight changes in curve derivative in the right plots (these change does not appear in relative time). The bottom plots figure the value of the continuous variable $$y (the same changes in the tempo are applied) defined by:

```
Curve @grain 0.2s { $$y { {0} 6 {6} } }
```

Despite the specification of the curve sampling rate (used within the reactive engine), the continuous control variable samples the line every $1/44100 = 0.022$ ms during audio processing.

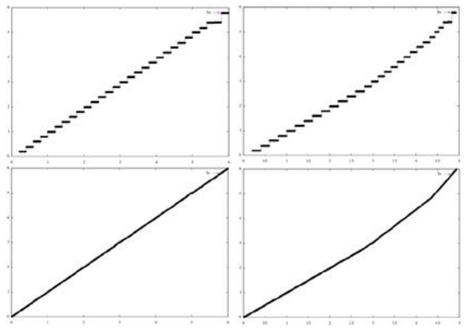


Figure 8. *Top:* plot of the values of the variable $y in relative and absolute time. There is 3 changes in the tempo during a linear ramp. *Bottom:* plot of the value of the continuous variable $$y. The same changes in the tempo are applied.

4. ANTHÈME 2 EXAMPLE

We validated the previous extension in a a strict re-implementation of *Anthème 2* introduced in section 2. For this reimplementation, we aim at embedding all audio computation inside an *Antescofo* program and next to control parameters following extensions. Code listing 2 shows a message-passing implementation where level control and DSP parameter (frequency shift value here) are passed to an outside module corresponding to implementation in section 2; and Code listing 3 shows the new implementation with strictly the same behavior. In Code 3, the level parameter is defined by a Curve control construct, and the DSP node employs faust :: PitchShifter . The definition of PitchShifter is a native *Faust* code included in the same scope, and is compiled on-the-fly upon score load in *Antescofo*. Computation of PitchShifter is handled dynamically during runtime executation of the system (i.e. live music performance) and parameters combine discrete values ($freq), interpolated variables ($psout) as well as audio buffer input ($$audioIn) and an audio buffer output link $$linkFS which is sent later on for spatial panning. Implementation of other modules follow the same procedure, by embedding native *Faust* code (for time-domain signal processing).

```
TRILL ( 8100 8200 ) 7/3 Q25
  ; bring level up to 0db in
      25ms
  fs-out-db 0.0 25
  ; frequency shift value
  fd1_fre -205.0
```

Code 2. Message Passing (old style)

```
TRILL ( 8100 8200 ) 7/3 Q25
  Curve c3 @grain := 1ms
  {; bring level up to 0db in
      25ms
    $psout
    {
      {0}
      25ms {1}
    }
  }

  $freq := -205 ; freq.
      shift value
  $$linkFS := faust::
      PitchShifter(
      $$audioIn, $freq,
      $psout)
```

Code 3. Embedded Audio

Time-profiling analysis of the message-passing example (Figure 1) and its embedded counter-part shows an improvement of 12% overall system utility improvement with the new implementation, corresponding to 46% utility performance on the task itself. This analysis was done on a MacBook and using *XCode*'s *Time Profiler* tool on a sampled period of real-time performance simulations where the code interacts with a human musician.

This improvement is due to several factors: optimisation of local DSP code provided by native hosts (such as *Faust*) and the lazy type conversion approach adopted in section 3 when converting (for example) between Curve and continuous-audio variables.

The approach developped here provides some gain in performance but also preserve both the control structure of the program for designers and its final behavior. This is made possible by explicit consideration of timing control over computations at stake and embedding them into the coordination system of *Antescofo*. The performance improvement has also allowed us to prototype such interactive music pieces on mini-computers such as Raspberry PI and UDOO.

5. CONCLUSION

We extended the *Antescofo* language, already employed by the community in various creations involving musicians and computers worldwide, with the possibility of time-aware combination of signal processing and control during authorship and real-time performance. This is achieved through a *GALS* execution model as shown in section 3, embedding of existing local modules in their native language in the system and compiled on the fly, and assuring their timely inter-communication and constraints inherited from their types or computation at stake. We showcased this study by extending an existing implementation of a music piece in the general repertoire (namely, Pierre Boulez' "Anthèmes 2") using the proposed approach. We showed its potential both for behavior preservation, time precision, ease of programming without significant breakdown for designers, and potential for multi-platform deployment.

This work will be extended in several ways, providing more native embedding services to users based on existing practices in interactive multimedia. The type system can be enriched for adequate description of finer temporal relationships. More studies and benchmark should be undertaken when combining and deploying multiple-rate processing modules and their synchrony in the system. Static Analysis tools should provide feedback to designers when certain timing constraints between computational modules can not be held, as an extension to [11], and could enable further optimizations in the DSP graph, and in the interaction between signal processing and control. The listening module of Antescofo could be reimplemented as an audio effect, opening the way to various specialized listening modules.

Acknowledgments

This work was partially funded by the French National Research Agency (ANR) INEDIT Project (ANR-12-CORD-0009) and the INRIA internship program with Chile.

6. REFERENCES

[1] G. Berry and G. Gonthier, "The Esterel Synchronous Programming Language: Design, Semantics, Implementation," *Sci. Comput. Program.*, vol. 19, no. 2, pp. 87–152, 1992.

[2] A. Cont, "Antescofo: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music," in *Proceedings of International Computer Music Conference (ICMC)*, Belfast, Irlande du Nord, August 2008.

[3] ——, "A Coupled Duration-Focused Architecture for Real-Time Music-to-Score Alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 974–987, 2010.

[4] A. Cont, J. Echeveste, J.-L. Giavitto, and F. Jacquemard, "Correct Automatic Accompaniment Despite Machine Listening or Human Errors in Antescofo," in *Proceedings of International Computer Music Conference (ICMC)*. Ljubljana, Slovenia: IRZU - the Institute for Sonic Arts Research, Sep. 2012. [Online]. Available: http://hal.inria.fr/hal-00718854

[5] A. Cont and J.-L. Giavitto, "Antescofo workshop at ICMC: Composing and performing with antescofo," in *Joint ICMC - SMC Conference*, Athens, Greece, Sep. 2014, The remake of *Anthèmes 2* is part of the tutorial and it can be downloaded at http://forumnet.ircam.fr/products/antescofo/.

[6] F. Doucet, M. Menarini, I. H. Krüger, R. Gupta, and J.-P. Talpin, "A verification approach for gals integration of synchronous components," *Electronic Notes in Theoretical Computer Science*, vol. 146, no. 2, pp. 105–131, 2006.

[7] J. Echeveste, J.-L. Giavitto, and A. Cont, "Programming with Events and Durations in Multiple Times: The Antescofo DSL," *ACM Trans. on Programming Languages and Systems (TOPLAS)*, 2015, (submitted).

[8] N. Halbwachs, *Synchronous Programming of Reactive Systems*, ser. Lecture Notes in Computer Science, A. J. Hu and M. Y. Vardi, Eds. Springer, 1998, vol. 1427.

[9] E. A. Lee, "Computing Needs Time," *Communications of the ACM*, vol. 52, no. 5, pp. 70–79, 2009.

[10] Y. Orlarey, D. Fober, and S. Letz, *FAUST : an Efficient Functional Approach to DSP Programming*, 2009, pp. 65–96. [Online]. Available: http://www.grame.fr/ressources/publications/faust-chapter.pdf

[11] C. Poncelet and F. Jacquemard, "Model based testing of an interactive music system," in *ACM SAC*, 2015.

[12] M. Puckette, "Pure data," in *Proc. Int. Computer Music Conf.*, Thessaloniki, Greece, September 1997, pp. 224–227. [Online]. Available: http://www.crca.ucsd.edu/~msp

[13] ——, "Combining Event and Signal Processing in the MAX Graphical Programming Environment," in *Proceedings of International Computer Music Conference (ICMC)*, vol. 15, Montréal, Canada, 1991, pp. 68–77.

[14] R. V. Rasmussen and M. A. Trick, "Round robin scheduling – a survey," *European Journal of Operational Research*, vol. 188, no. 3, pp. 617 – 636, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221707005309

[15] R. Rowe, *Interactive music systems: machine listening and composing*. Cambridge, MA, USA: MIT Press, 1992.

[16] A. Sorensen and H. Gardner, "Programming With Time: Cyber-Physical Programming With Impromptu," in *ACM Sigplan Notices*, vol. 45, no. 10. ACM, 2010, pp. 822–834.

[17] G. Wang, P. R. Cook, and S. Salazar, "Chuck: A strongly timed computer music language," *Computer Music Journal*, 2016.

A Review of Interactive Conducting Systems: 1970-2015

Kyungho Lee, Michael J. Junokas, Guy E. Garnett
Illinois Informatics Institute
University of Illinois at Urbana-Champaign
1205 W. Clark NCSA Building, Urbana, IL USA
klee141, junokas, garnett@illinois.edu

ABSTRACT

Inspired by the expressiveness of gestures used by conductors, research in designing interactive conducting systems has explored numerous techniques. The design of more natural, expressive, and intuitive interfaces for communicating with computers could benefit from such techniques. The growth of whole-body interaction systems using motion-capture sensors creates enormous incentives for better understanding this research. To that end, we retraced the history of interactive conducting systems that attempt to come to grips with interpreting and exploiting the full potential of expressivity in the movement of conductors and to apply that to a computer interface. We focused on 55 papers, published from 1970 to 2015, that form the core of this history. We examined each system using four categories: interface (hardware), gestures (features), computational methods, and output parameters. We then conducted a thematic analysis, discussing how insights have inspired researchers to design a better user experience, improving naturalness, expressiveness and intuitiveness in interfaces over four decades.

1. INTRODUCTION

In the history of Western art music, conductors have served as both physical and conceptual focal points. The modern form of conducting emerged due to the increasing complexity of symphonic scores over the nineteenth century. They became fully-fledged members of the performing ensemble, generating a stream of musical expression "running from composer to individual listener through the medium of the performer and further mediated by the expressive motions of the conductor." [1] In order to accomplish this goal, they used a variety of physical signatures to seamlessly convey musical expressions to the ensemble throughout rehearsals and performances. Conductors, in their increasingly complex task of directing the orchestra, have increasingly learned how to use embodied knowledge, as musicians and dancers did before them. Recent research supports this concept, showing that as a series of emblematic gestures, conducting has the capability of transmitting specific musical ideas, using a wide range of physical expressivity [2] [3].

With recent advances in sensing technology, the potential use of whole-body interaction (WBI) [4] plays a pivotal role in enhancing the natural user-interaction (NUI) paradigm, with an emphasis on embodiment. Since the field of WBI or NUI is relatively young and finds a novel interaction model to move researchers forward, conducting gestures have attracted researchers who seek fundamental insight into the design of complex, expressive, and multimodal interfaces. While the current natural-user interaction design paradigm has the ability to recognize the user's gestures and to operate a set of commands, it is still limited in extracting the expressive content from gesture, and even more limited in its ability to use this to drive an interactive system. The design of conducting interfaces has been driven by new methods or models that empower users through the augmentation of expression and/or expanding to a new degree of control to challenge the limitations. Our motivation is to start a systemic review of the history and state of the art, derived from these questions: What are the significant documents and experiments in the development of conducting systems? What is the research history and legacy of this domain? What can we learn from this body of research that might help us to design a better user experience?

Our paper will address interfaces that have been designed to capture conducting gestures, features and computational methods that have been used to interpret expressive contents in gestures, and strategies and techniques that have been used to define effective mappings from gesture to control of sound. Based on these points, we conducted a systematic review of fifty-five papers that used conducting gestures in interactive system design. This review is comprised of a sub-sample of papers related to interactive conducting systems that were selected from a broader literature search exploring the impact of designing multi-modal, expressive interfaces. A narrative review was additionally carried out in order to develop a more coherent understanding of expression-driven gesture design that supports human creativity, focusing on translating musical expression using the gesture. From this range of papers, three major themes in the history of designing interfaces with conducting gestures were addressed: naturalness, intuitiveness, and expressiveness. We described the keywords in the implications section in detail.

2. TERMINOLOGY

In this section, we present consensus-derived, fundamental concepts and definitions of interactive conducting systems, providing readers with a fundamental background for the

better understanding the rest of the paper.

2.1 Interactive Conducting Systems

By referring to interactive 'conducting' systems, our focus narrows to a subset of interactive systems that use the breadth of standard, or typical, conducting gestures. Different researchers have defined the term in different ways. Early pioneers, for example, defined their systems as a conducting system [5], a music system, a conducting program [6], and a conductor follower [7]. In this paper, we define the term, *interactive conducting system*, as a system that is able to capture gestures from a conductor (as a user), extrapolate expressive contents in the gestures, assign appropriate meaning, and apply that meaning to the control of sound or other output media. Using such gestures, the conductor can manipulate a set of parameters interpreted by the system to produce outputs such as MIDI note/score playbacks, sound waveforms, and/or visual elements according to prescribed mapping strategies.

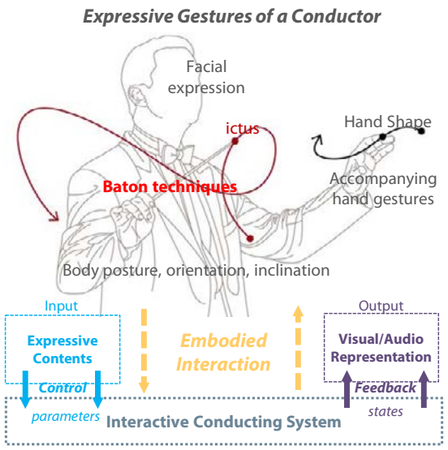


Figure 1. Illustration of an interactive conducting system, showing how conductors can drive a system using conducting gestures.

Figure 1 illustrates how an interactive conducting system works from our perspective. Note that the term 'embodied interaction' refers to using the perceivable, actionable, and bodily-experienced (embodied) knowledge of the user in the proximate environment (interactive system).

2.2 Conducting Gestures and Expressivity

A conductor uses expressive gestures to shape the musical parameters of a performance, interacting with the orchestra to realize the desired musical interpretation. While a conductor is directing, he or she makes use of diverse, often idiosyncratic, physical signatures such as facial expression, arm movement, body posture, and hand shape as seen in Figure 1. These physical signatures can convey different types of information simultaneously. Amongst these four different types of information channels, researchers have been mainly interested in the use of the hand and arm gestures in referring to conducting gestures, largely because these are the most standardized elements of the technique. Theoretically, conducting gestures have been investigated in linguistics as emblematic and pantomimic gestures according to the spectrum of the Kendon's continuum [8].

Based on this theoretical background, conducting gestures can be understood as a stream of linguistic information, which is relatively fixed, and lexicalized. There is very little variety in conveying specific musical direction to the others and decoding it from the gestures [9]. This view was addressed in Max Rudolf's authoritative conducting textbook [10] where he defined explicit parts of 'conducting gestures.' For example, conducting gestures can be classified into several groups by their intended effect (musical information) on the performance as done with baton techniques, which have been used to indicate the expression of each beat (e.g., legato, staccato, marcato, and tenuto), while accompanying left hand gestures have been used to support controlling dynamics, cues, cutoffs and vice versa.

There is a novel interpretation where the degree of variation in conducting gestures is used to enhance expression from the HCI perspective. Different conductors might perform the same musical expressions differently under the grammar of conducting. Therefore, we can consider expressivity is more associated with how to perform rather than what to perform.' Recent empirical research outcomes claimed that the individual variance or gestural differentiation can be understood as a degree of expression [2] providing a rich research area. Similarly, Caramiaux et al. [11] claimed that such differentiation can add "meaningful variation in the execution of a gesture" in expression-oriented interactions.

3. METHODOLOGY

3.1 Planning the Review

In this section, we identified the need for a systematic literature review, and developed a protocol that specifies methods to conduct data collection and analysis.

- Objective: Analyzing interactive conducting systems and computational methods used to find the challenges and opportunities for designing a better interactive systems, enabling the of use expressive, multimodal inputs.

- Research questions: (1) What types of interfaces have been designed to capture conducting gestures? (2) What features and computational methods have been applied to interpret expressive contents in gestures? (3) What strategies and techniques have been used to create effective mappings between these input gestures and applied outputs?

- Research sources: ACM, IEEE, CiteSeerX, SpringerLink, Computer Music Journal, Journal of New Music Research.

- Search strings: Our primary objective focuses on capturing and extrapolating expressivity from the conducting gestures, so we chose the following search strings after preliminary searches across the disciplines of musicology, psychology, machine learning, pattern recognition, and HCI studies. The first search string focuses on the design of interactive system that using conducting gestures. The second focuses on gesture recognition and analysis of conducting gestures using computational methods. The third focuses on the application of conducting gestures and movement.

  1. conductor **and** (gesture or movement) **and** (interface **or** system) **or** (orchestra **or** ensemble) 2. conducting gesture **and** (expression **or** expressive gestures) **or** (recognition **or** analysis) 3. conducting gesture **and** (expression **or** expressive gestures) **or** (visual **or** sound)

- Language/Time restriction: Any papers published in English and available in the digital library.

- Inclusion criteria: (I1) Research comprising strategies, methods and techniques for capturing conducting gesture (conductor's gesture) and applying the results to design an interactive system/interface; (I2) Studies comprising theoretical backgrounds and computational methods to analyze and recognize characteristic aspects of conducting gestures; (I3) Projects using conducting gestures or conductors? expressions to drive interactive system to generate visuals/ audio.

- Exclusion criteria: (E1) Studies which do not meet any inclusion criteria; (E2) Studies focusing on conducting gestures using a computational approach but not related to any design aspect of HCI; (E3) Studies focusing on qualitative analysis of conducting gestures but not providing any computational methods; (E4) If two papers, from the same authors published in the same year, cover the same scope, the older one was excluded.

3.2 Conducting the Review

After defining a review protocol, we conducted the review. The data collection started at the beginning of 2015 with initial searches returning 129 studies with some overlapping results among the sources. After applying the inclusion, exclusion, and quality criteria, 55 papers were selected. The papers were primarily collected from ICMC (19 papers), ACM (4), IEEE (3), Computer Music Journal (2) and Journal of New Music Research (2). Other sources were collected from university data repositories (dissertation/thesis) or other journals.

4. RESULTS

Based on our investigation, we developed six different themes that have been centered around the history of interactive conducting systems: pioneers; tangible user interface; gesture recognition/machine learning; sound synthesis; commercial sensors; and visualization.

4.1 The first interactive conducting systems

Early interactive conducting systems design resorted to control and interaction paradigms of the time. They incorporated knobs, 3D joysticks, and keyboards as input devices. However, a series of pioneering explorations considering Engelbart's seminal demo [12] was presented in 1968. Mathews [13] described that his desire was to create an interface that would be able to connect the computer to the user as a conductor is connected to the orchestra. He fed the score information to the computer, which was paired with user interactions to make dynamic score interactions. Also, he adopted three modes (the score, rehearsal, and performance) that reflected the mental model of conductors. The name, a conducting system, was explicitly entitled by Buxton later in 1980. In Buxton et al.'s work [5], improved design considerations in terms of graphical representation were implemented. Such considerations enabled the user to adjust various musical parameters, such as tempo, articulation, amplitude, richness(timbre), on the screen through a textual user interface. The user controlled the parameters by typing numbers or moving cursors. These systems explored the potential of using non-conventional modalities and demonstrated how interactive conducting systems were being developed.

4.2 Rise of tangible user interface

Tangible interaction design generally encompasses user interfaces and interactions that emphasize "materiality of the interface; physical embodiment; whole-body interaction; the embedding of the interface and the users' interaction in real spaces and contexts." [14] Although this period was right before the explosion of tangible user interface design, we can see researchers' design reflecting its philosophy. From 1979, Mathews and Abbott [15] started designing a mechanical baton to use as an input device, allowing users to provide more intuitive input through its use. The baton was struck by the user with his or her hands or sticks and required no prior training for use. This tangible interface provided the user with the ability to capture the mental model of a conductor through the use of his or her embodied interaction with the machine. The consideration of tangibility and intuitiveness was advanced further by Keane et al. [16] starting in 1989. They designed a wired baton, which resembled an ordinary baton but was augmented with spring wires and an metal ball inside. By 1991, they improved the MIDI baton by adding a wireless transmitter and expanding the number of MIDI channels to 16, allowing the control of multiple parameters at the same time. Marrin et. al's Conductor's Jacket [17] further expanded this interface category. It is a wearable device that demonstrates the potential power of using EMG sensors, attempting to map expressive features to sections in the music score. Due to technological limits of this period, the overall weight of the device, including the digital baton, was a potential concern. In her later project, You're the Conductor [18] and Virtual Maestro [19], Marrin and her collaborators developed a gesture recognition system that was capable of mapping the velocity and the size of gestures to musical tempos and dynamics. Her approach has inspired numerous researchers interested in using the Arduino and accelerometers to measure the body's movement.

4.3 Use of Machine Learning Approach

In the history of interactive conducting systems, there have been three main challenges related with machine learn-

ing (ML): data collection, feature generation, and modeling. The first challenge was collecting conducting gestures and assuring the quality of this gestural dataset by removing outliers and smoothing signals. Many researchers needed to implement physical interfaces to measure the user's movement with higher precision. The second challenge was to find reliable and discriminative features to extrapolate expressivity from gestures including a dimensionality reduction process. A great deal of research has adopted kinematic features such as the velocity and acceleration to describe the movement. The third challenge was modeling the temporal dynamics of conducting gestures. Researchers have used Hidden Markov models (HMM) or neural networks (ANN) to create such models. Bien et al.'s work [20] was one of the first to adopt fuzzy logic to capture the trajectory of a baton in order to determine the beat. However, they built it based on the IF-THEN rule-based fuzzy system, not fully exploiting the potentials of fuzzy logic might have. Lee M. [7], Brecht [21] and their colleagues brought ANN to address conducting gesture recognition, using the Buchla Lightning baton [22] as an input device. They trained a two-layer multi perceptron (MLP) between six different marker points, time, and the probability of the next beat, using the ANN was adopted to deal with the local variations in conducting curves. Sawada et al. [23] [24] and Usa [25] also used ANN and HMM in their works respectively. In 2001, Garnett and his colleagues [26] advanced the algorithm by using distributed computing via open sound control(OSC), building on the success of the conductor follower. Kolesnik and Wanderley [27] proposed a system that captured conducting gestures using a pair of cameras, analyzing the images using EyesWeb. They used an HMM to recognize the beat and amplitude from the right and left hand expressive gestures. The exploration of ML approaches was accelerated by the advent of commercial sensors such as the Nintendo's Wiimote and the Microsoft's depth sensor, the Kinect V1 and V2. Bradshaw and Ng [28] adopted the Wiimote to analyze conducting gestures whereas other researchers [29] [30] [31] used the Kinect as an input sensor. Dansereau et al. [32] captured baton trajectories using a high quality motion capture devices (Vicon) and analyzed them by applying an extended Kalman filter as a smoothing method, using a particle filter for a training. Although the capability of capturing conducting gestures was advanced over time, the tracking results suggested that there were a lack of advancements in the input-output mappings, maintaining basic output parameters such as beat pattern, dynamics, and volume.

4.4 Sound Synthesis

One of the pioneering projects, GROOVE [13], was designed for "creating, storing, reproducing, and editing functions of time," for sound synthesis. After that, many researchers put their efforts into developing systems that enabled the user to control musical parameters in MIDI scores and audio files. Their projects allowed users to directly manipulate musical performances, mapping kinetic movements to sound. Morita et al. [33] began realizing a system that gave "an improvisational performance in real-time." To achieve their goal, they adopted computer vision technology to track the conductor's baton. With the system, the user can manipulate tempo, strength (velocity), start, and stop of the music. In following work, they extended the system, adding a data glove to capture additional expressions of hand shapes. From 2001, Borchers et al. [34] presented a series of "personal orchestra" projects, allowing the user to control tempo, dynamics, and instrument emphasis based on pre-recorded audio files. During the same period, Murphy et al. [35] and Kolesnik [27] attempted to implement systems to play time-stretched sound in real time using a variant of the phase vocoder algorithm. However, computing power was not sufficient to guarantee synchronous audio and video playback, so video or audio playback module were dealt with independently. In this regard, Lee and his colleagues' work had significantly contributed to addressing these problems. He described his concept as semantic time [36], aiming to allow the user to perform time-stretching without substantially losing or distorting the original information. He applied the technique to multiple projects: *conga*, *You're conductor* and *iSymphony* [37] [18]

4.5 Advent of commercial sensors

Until the 2000's, many researchers investigated a conductor's gestures by attaching customized sensors to body parts or analyzing motion in a lab context to acquire the highest quality of datasets. However, the advent of relatively cheap and robust sensors, such as Nintendo Wiimote and Microsoft Kinect, led researchers to a different approach. Nintendo introduced the Wiimote in late 2006 as an advanced input device incorporating a 3-axes accelerometer and infrared sensor. It supported the Bluetooth protocol for communication. Microsoft Kinect, which was presented in 2009 for V1 and 2014 for V2, was featured an RGB camera, depth sensor, and a microphone array. One of the primary reason for adopting commercial sensors is that they are less expensive, non-invasive, yet powerful and can be used in general contexts which accelerates the data collection and iterative design process. By the year of 2000, many research projects had been designed to use these sensors. Bradshaw and Ng [28] used multiple Wiimotes to capture 3D acceleration data of conducting gestures. They attempted to extract information and use the parameters to change tempo and dynamic then feed them back to the user using several appropriate methods including sonification, visualization and haptics (i.e vibration in the controller). Toh et al. [29] designed an interactive conducting system using the Kinect V1, allowing the user to control tempo, volume, and instrument emphasis. It was also one of the first attempts at using a body posture for control information. Rosa et al. [30] designed another system that allowed the user to conduct a virtual orchestra, controlling the tempo, the overall dynamics, and the specific volume levels for sets of instruments in the orchestra.

4.6 Visualization of expressivity

Unlike the other advancements in the history of designing interactive conducting systems, little attention has been paid to visualizing the dynamics of conducting gestures and its expressivity. The uncharted territory is challenging due to: 1) the concrete conceptual model that lead researchers to understand the qualitative aspect of conducting gestures. 2) the feature generation and recognition methods to analyze and extract expressivity from the movements. Nevertheless, there were several attempts to visualize some dimensions of conducting gestures. One of the early attempts was made in Garnett et al.'s project [38], Virtual Conducting Practice Environment. They visualized the four beats in 4/4 beat pattern and the horizontal line representing the beat plane. In 2000, Segen and Gluckman [39] presented their project, Visual Interface for Conducting Virtual Orchestra, at SIGGRAPH. While the MIDI sequencer was playing an orchestral score, the user was able to adjust its tempo and volume. 3D human models were rendered and animated, that follow pre-designed movements and choreography, based on the tempo set. Bos et al. [40] implemented the virtual conductor system that conducted music specified by a MIDI file to human performers. It received input from a microphone, responding to the tempo of the musicians. This was the first use of a virtual agent to direct other human agents instead of being controlled by the user. Recently, Lee et al. [41] created an interactive visualization to represent expressivity of the conducting gestures. They adopted Laban Movement Analysis to parameterize expressivity. The visualization received an input video stream and was driven by expressive motion parameters extracted from the user gestures, rendering particle graphics.

5. IMPLICATIONS

Based on the synthesis of the survey, we drew three implications for future design works. These implications reflect the current trend of designing WBI/NUI paradigm based on Norman and van Dam's note. Norman proposed that designers could improve user performance by mapping knowledge in the world to expected knowledge in the user's mind [42]. van Dam suggested that the ideal user interface "would let us perform our tasks without being aware of the interface as the intermediary." [43] Upon consideration, the future of interactive conducting systems should consider the three core elements one step further: 1) *naturalness* which is allowing a multi-limbed and multi-modal interaction; 2) *intuitiveness* which is enabling embodied interaction; 3) *expressiveness* which is inspiring the user's creative tasks through transmodal feedbacks. We describe each implication in more detail.

5.1 For Being Natural

Amongst several definitions, we can define being natural in our context as a sensing technique for having more holistic forms of inputs that allow the user to use multi-limbed and multi-modal interaction. With advanced sensing mechanisms, we witnessed that new forms of 'natural' input have arisen to replace traditional WIMP based mechanisms. With machine learning techniques, the whole-body interaction can make the best use of our embodied abilities and real world knowledge [44]. However, our analysis results suggest that we need to explore other techniques to extrapolate expressivity in conducting gestures revealed not only through movement, but also through facial expressions, muscles tensions, or brain activities. Because current models and sensors are not sensitive enough to extrapolate affective or cognitive states from subtle gestures (external cues) that represent the internal cognitive or affective state indirectly [45]. In addition to sensing external cues, we can consider adopting Brain-Computer Interfaces (BCI) to capture significant insight from the users' emotional state more directly. By adopting BCIs, we can utilize rich information not only to operate a set of commands with the user's brain activity instead of using motor movements but also to provide more natural ways of controlling interfaces. For example, recalling a pleasant moment could be recognized and interpreted as expression parameters to control the system in the highest possible natural and intuitive manner.

5.2 For Being Intuitive

Raskin [46] argued that an 'intuitive' interactive system should work in a similar way that the user does without pre-training or rational thought. He suggested that a user interface could incorporate intuitiveness by designing towards (even identically) something the user already knows. In the history of the interactive conducting systems, numerous researchers have designed tangible interfaces and created visualizations that resembled the real-world context of conductors to keep their mental model as similar as possible developed under the term of intuitive design. We propose to put more consideration on embodied interaction in the design process. A growing body of research in the understanding body-mind linkages has supported this claim, explaining how abstract concepts and ideas can become closely tied to the bodily experiences of sensations and movements. In the HCI fields, Höök [47] provided evidence of how "our corporeal bodies in interaction can create strong affective experiences." It is expected that the embodied interaction design approach will improve the overall user experience and the performance of conducting machines. As Norman [48] noted, designers can improve user performance with the interactive system by providing a better mapping knowledge from the world (determined by system design) to expected knowledge in the user's head.

5.3 For Being Expressive

Dobrian claims [49], musical instruments or interfaces cannot be expressive as they do not have anything to express until the user commands what to express and how to express it. However, we observed a great deal of ideas utilizing computers as a vehicle to transmit a conductor's expressiveness to the machine and to the audience in the history. Researchers have explored a variety of ways to quantify conductors' gesture and to transform the significance of expressivity into a mental musical representation. The exploration can be interpreted as a journey of designing creativity support tools in the music domain as we saw with many researchers experimenting with scores composed in a MIDI or waveforms producing the different quality of sound in their evaluation process. Our analysis demonstrated that only very few visual explorations were made through the history of interactive conducting, and further exploration is rich with opportunity. In this context, the concept of metacognition gives us evidence to consider adoption since it explains how our cognitive system evaluates and monitors our own thinking processes and knowl-

edge content [50]. Research findings showed that the metacognitive feeling of knowing, so-called confidence, can help the users to associate possible ideas together, guiding the users to a path to accomplish the goal [51].

6. CONCLUSION AND FUTURE WORK

We found that numerous interactive conducting systems had been researched and implemented over forty years reflecting the emerging technologies and paradigms from the HCI. Interactive conducting systems explore numerous, different approaches to making the best use of expressivity in conducting gestures from different perspectives; the kinematics of conducting gestures associated with tracking beats; the recognition of particular types of conducting gestures including articulation styles; and mapping for music control or synthesis. The interactive conducting systems were also developed and evaluated for various purposes such as performance, pedagogy, and scientific research prototypes to validate theory or algorithms. With three design implications, we can imagine the possible interactive system scenarios such as: 1) 'a machine symphony' which enables the conductors (the users) to lead a full-size orchestra made of 70-100 high quality virtual instruments based on MIDI scores; 2) 'an augmented ensemble' which visualizes expressivity in the conductors' movement through augmented/mixed reality technology; and 3) 'a pedagogical agent' that helps the users' embodied learning process for basic components of conducting gestures such as beat patterns and articulations styles.

Acknowledgments

This research was supported by the Social Sciences and Humanities Research Council of Canada (SSHRC).

7. REFERENCES

[1] C. Small, "Musicking–the meanings of performing and listening," vol. 1, no. 1, p. 9.

[2] G. Luck, P. Toiviainen, and M. R. Thompson, "Perception of expression in conductors' gestures: A continuous response study."

[3] G. D. Sousa, "Musical conducting emblems: An investigation of the use of specific conducting gestures by instrumental conductors and their interpretation by instrumental performers," Ph.D. dissertation, The Ohio State University, 1988.

[4] D. England, M. Randles, P. Fergus, and A. Taleb-Bendiab, "Towards an advanced framework for whole body interaction," in *Virtual and Mixed Reality*. Springer, pp. 32–40.

[5] W. Buxton, W. Reeves, G. Fedorkow, K. C. Smith, and R. Baecker, "A microcomputer-based conducting system," pp. 8–21.

[6] R. B. Dannenberg and K. Bookstein, "Practical Aspects of a midi conducting program," in *Proceedings of the 1991 International Computer Music Conference*, pp. 537–40.

[7] M. Lee, G. Garnett, and D. Wessel, "An adaptive conductor follower," in *Proceedings of the International Computer Music Conference*. International Computer Music Association, pp. 454–454.

[8] D. McNeill, *Gesture and thought*. University of Chicago Press.

[9] A. Gritten and E. King, Eds., *New perspectives on music and gesture*, ser. SEMPRE studies in the psychology of music. Farnham ; Burlington, VT: Ashgate Pub, 2011.

[10] M. Rudolf, *The grammar of conducting: a practical guide to baton technique and orchestral interpretation*. Schirmer Books.

[11] B. Caramiaux, M. Donnarumma, and A. Tanaka, "Understanding Gesture Expressivity through Muscle Sensing," vol. 21, no. 6, p. 31.

[12] D. C. Engelbart and W. K. English, "A research center for augmenting human intellect," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. ACM, 1968, pp. 395–410.

[13] M. V. Mathews and F. R. Moore, "GROOVE—a program to compose, store, and edit functions of time," vol. 13, no. 12, pp. 715–721.

[14] B. Ullmer and H. Ishii, "Emerging frameworks for tangible user interfaces," *IBM systems journal*, vol. 39, no. 3.4, pp. 915–931, 2000.

[15] M. V. Mathews and C. Abbott, "The sequential drum," pp. 45–59.

[16] D. Keane, *The MIDI baton*. Ann Arbor, MI: MPublishing, University of Michigan Library.

[17] T. Marrin and R. Picard, "The 'Conductor's Jacket': A Device for Recording Expressive Musical Gestures," in *Proceedings of the International Computer Music Conference*. Citeseer, 1998, pp. 215–219.

[18] E. Lee, T. M. Nakra, and J. Borchers, "You'Re the Conductor: A Realistic Interactive Conducting System for Children," in *Proceedings of the 2004 Conference on New Interfaces for Musical Expression*, ser. NIME '04. Singapore, Singapore: National University of Singapore, 2004, pp. 68–73.

[19] T. M. Nakra, Y. Ivanov, P. Smaragdis, and C. Ault, "The ubs virtual maestro: An interactive conducting system," pp. 250–255.

[20] Z. Bien and J.-S. Kim, "On-line analysis of music conductor's two-dimensional motion," in *, IEEE International Conference on Fuzzy Systems, 1992*, pp. 1047–1053.

[21] B. Brecht and G. Garnett, "Conductor Follower," in *ICMC Proceedings*, 1995, pp. 185–186.

[22] R. Rich, "Buchla Lightning MIDI Controller: A Powerful New MIDI Controller is Nothing to Shake a Stick at," *Electron. Music.*, vol. 7, no. 10, pp. 102–108, Oct. 1991.

[23] H. Sawada, S. Ohkura, and S. Hashimoto, "Gesture analysis using 3D acceleration sensor for music control," in *Proc. Int'l Computer Music Conf.(ICMC 95)*.

[24] T. Ilmonen and T. Takala, *Conductor Following With Artificial Neural Networks*.

[25] S. Usa and Y. Mochida, "A conducting recognition system on the model of musicians' process," vol. 19, no. 4, pp. 275–287.

[26] G. E. Garnett, M. Jonnalagadda, I. Elezovic, T. Johnson, and K. Small, "Technological advances for conducting a virtual ensemble," in *International Computer Music Conference,(Habana, Cuba, 2001)*, pp. 167–169.

[27] P. Kolesnik and M. Wanderley, "Recognition, analysis and performance with expressive conducting gestures," in *Proceedings of the International Computer Music Conference*, pp. 572–575.

[28] D. Bradshaw and K. Ng, "Analyzing a conductors gestures with the Wiimote," pp. 22–24.

[29] L. Toh, W. Chao, and Y.-S. Chen, "An interactive conducting system using Kinect," 2013, dOI: 10.1109/ICME.2013.6607481.

[30] A. Rosa-Pujazon, I. Barbancho, L. J. Tardon, and A. M. Barbancho, "Conducting a virtual ensemble with a kinect device," in *Proceedings of the Sound and Music Computing Conference 2013*, ser. SMC'13. Logos Verlag Berlin, pp. 284–291.

[31] Á. Sarasúa and E. Guaus, "Dynamics in music conducting: A computational comparative study among subjects," in *14th International conference on New interfaces for musical expression*, ser. NIME'14, vol. 14.

[32] D. G. Dansereau, N. Brock, and J. R. Cooperstock, "Predicting an orchestral conductor's baton movements using machine learning," vol. 37, no. 2, pp. 28–45.

[33] H. Morita, S. Hashimoto, and S. Ohteru, "A computer music system that follows a human conductor," vol. 24, no. 7, pp. 44–53.

[34] J. O. Borchers, W. Samminger, and M. Mühlhäuser, "Conducting a realistic electronic orchestra," in *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, pp. 161–162.

[35] D. Murphy, T. H. Andersen, and K. Jensen, "Conducting audio files via computer vision," in *Gesture-based communication in human-computer interaction*. Springer, pp. 529–540.

[36] E. Lee, T. Karrer, and J. Borchers, "Toward a framework for interactive systems to conduct digital audio and video streams," *Computer Music Journal*, vol. 30, no. 1, pp. 21–36, 2006.

[37] E. Lee, I. Grüll, H. Kiel, and J. Borchers, "conga: A framework for adaptive conducting gesture analysis," in *Proceedings of the 2006 conference on New interfaces for musical expression*. IRCAM—Centre Pompidou, pp. 260–265.

[38] G. E. Garnett, F. Malvar-Ruiz, and F. Stoltzfus, "Virtual conducting practice environment," in *Proceedings of the International Computer Music Conference*, pp. 371–374.

[39] J. Segen, J. Gluckman, and S. Kumar, "Visual interface for conducting virtual orchestra," in *15th International Conference on Pattern Recognition, 2000. Proceedings*, vol. 1, pp. 276–279 vol.1.

[40] P. Bos, D. Reidsma, Z. Ruttkay, and A. Nijholt, "Interacting with a Virtual Conductor," in *Entertainment Computing - ICEC 2006*, ser. Lecture Notes in Computer Science, R. Harper, M. Rauterberg, and M. Combetto, Eds. Springer Berlin Heidelberg, no. 4161, pp. 25–30.

[41] K. Lee, D. J. Cox, G. E. Garnett, and M. J. Junokas, "Express It!: An Interactive System for Visualizing Expressiveness of Conductor's Gestures," in *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, ser. C&C '15. New York, NY, USA: ACM, 2015, pp. 141–150.

[42] D. A. Norman, *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[43] A. van Dam, "Beyond WIMP," *IEEE Computer Graphics and Applications*, vol. 20, no. 1, pp. 50–51, 2000.

[44] D. England, "Whole Body Interaction: An Introduction," in *Whole Body Interaction*, ser. Human-Computer Interaction Series, D. England, Ed. Springer London, pp. 1–5.

[45] D. Tan and A. Nijholt, "Brain-Computer Interfaces and Human-Computer Interaction," in *Brain-Computer Interfaces*, ser. Human-Computer Interaction Series, D. S. Tan and A. Nijholt, Eds. Springer London, pp. 3–19, DOI: 10.1007/978-1-84996-272-8_1.

[46] J. Raskin, "Intuitive equals familiar," vol. 37, no. 9, pp. 17+.

[47] K. Höök, "Affective loop experiences: designing for interactional embodiment," vol. 364, no. 1535, pp. 3585–3595.

[48] D. A. Norman, *The design of everyday things*, 1st ed. Doubleday,.

[49] C. Dobrian and D. Koppelman, "The'E'in NIME: musical expression with new computer interfaces," in *Proceedings of the 2006 conference on New interfaces for musical expression*. IRCAM—Centre Pompidou, pp. 277–282.

[50] C. Hertzog and D. F. Hultsch, "Metacognition in adulthood and old age."

[51] T. Bastick, *Intuition : how we think and act*. J. Wiley,.

O2: Rethinking Open Sound Control

Roger B. Dannenberg
Carnegie Mellon University
rbd@cs.cmu.edu

Zhang Chi
Carnegie Mellon and Tianjin University
zcdirk@gmail.com

ABSTRACT

O2 is a new communication protocol and implementation for music systems that aims to replace Open Sound Control (OSC). Many computer musicians routinely deal with problems of interconnection in local area networks, unreliable message delivery, and clock synchronization. O2 solves these problems, offering named services, automatic network address discovery, clock synchronization, and a reliable message delivery option, as well as interoperability with existing OSC libraries and applications. Aside from these new features, O2 owes much of its design to OSC and is mostly compatible with and similar to OSC. O2 addresses the problems of inter-process communication with a minimum of complexity.

1. INTRODUCTION

Music software and other artistic applications of computers are often organized as a collection of communicating processes. Simple protocols such as MIDI [7] and Open Sound Control (OSC) [1] have been very effective for this, allowing users to piece together systems in a modular fashion. Shared communication protocols allow implementers to use a variety of languages, apply off-the-shelf applications and devices, and interface with low-cost sensors and actuators. We introduce a new protocol, O2, in order to provide some important new features.

A common problem with existing protocols is initializing connections. For example, typical OSC servers do not have fixed IP addresses and cannot be found via DNS servers as is common with Web servers. Instead, OSC users usually enter IP addresses and port numbers manually. The numbers cannot be "compiled in" to code because IP addresses are dynamically assigned and could change between development, testing, and performance. O2 allows programmers to create and address services with fixed human-readable names.

Another desirable features is timed message deliveries. One powerful method of reducing timing jitter in networks is to pre-compute commands and send them in advance for precise delivery according to timestamps. O2 facilitates this forward synchronous approach [6] with timestamps and clocks.

Finally, music applications often have two conflicting requirements for message delivery. Sampled sensor data should be sent with minimum latency. Lost data is of little consequence since a new sensor reading will soon follow. This calls for a best-effort delivery mechanism such as UDP. On the other hand, some messages are critical, e.g. "stop now." These critical messages are best sent with a reliable delivery mechanism such as TCP.

Our goal has been to create a simple, extensible communication mechanism for modern computer music (and other) systems. O2 is inspired by OSC, but there are some important differences. While OSC does not specify details of the transport mechanism, O2 uses TCP and UDP over IP (which in turn can use Ethernet, WiFi, and other data link layers). By assuming a common IP transport layer, it is straightforward to add discovery, a reliable message option, and accurate timing.

In the following section, we describe O2, focusing on novel features. Section 3 presents related work. Then, in Sections 4 and 5, we describe the design and implementation, and in Section 6, we describe how O2 interoperates with other technologies. Section 7 describes our current implementation status, and a summary and conclusions are presented in Section 8.

2. O2 FEATURES AND API

The main organization of O2 is illustrated in Figure 1. Communication takes place between "services" which are addressed by name using an extension of OSC addressing in which the first node is considered a service name. For example, `"/synth/filter/cutoff"` might address a node in the `"synth"` service. To create a service, one writes

o2_initialize("`application`"); *// one-time startup*
o2_add_service("`service`"); *// per-service startup*
o2_add_method("`address`", "`types`", *handler, data*);

where "`application`" is an application name, used so that multiple O2 applications can co-exist on one network, and *o2_add_method* is called to install a handler for each node, where each "`address`" includes the service name as the first node.

Services are automatically detected and connected by O2. This solves the problem of manually entering IP addresses and port numbers. In addition, O2 runs a clock synchronization service to establish a shared clock across the distributed application. The master clock is provided to O2 by calling:

o2_set_clock(*clock_callback_fn*, *info_ptr*);

where *clock_callback_fn* is a function pointer that provides a time reference, and *info_ptr* is a parameter to pass to the function. The master clock can be the local system time of some host, an audio sample count converted to seconds (for synchronizing to audio), SMPTE time code, GPS, or any other time reference.

Messages can be sent either with lowest latency or reliably using two "flavors" of send function:

o2_send ("`address`", *time,* "`types`", val_1, val_2, …);
o2_send_cmd ("`address`", *time,* "`types`", val_1, val_2, …);

where "`types`" (in the C implementation) specifies the types of parameters, e.g. `"if"` means val_1 is an integer and val_2 is a float. The first form uses UDP, which is most common for OSC, and the second form sends a "command" using TCP, ensuring that the message will be delivered. Notice that every send command specifies a delivery *time*.

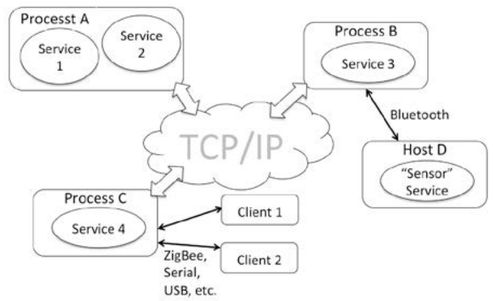


Figure 1. A distributed O2 application showing processes connected by TCP/IP (wireless and/or wired) over a local area network, running multiple services, with additional single-hop links over Bluetooth, ZigBee, etc. to both services and simple clients that do not receive messages. Services on Process A may run within a single process or in separate processes, and all processes may act as clients, sending messages to any service.

3. RELATED WORK

Open Sound Control (OSC) has been extremely successful as a communication protocol for a variety of music and media applications. The protocol is simple, extensible, and supported by many systems and implementations. The basic design supports a hierarchical address space of variables that can be set to typed values using messages. The messages can convey multiple values, and thus OSC may be viewed as a remote function or method invocation protocol. One very appealing quality of OSC, as compared to distributed object systems (such as CORBA [2]), is that OSC is very simple. In particular, the OSC address space is text-based and similar to a URL. It has been argued that OSC would be more efficient if it used fixed-length binary addresses, but OSC addresses are usually human-readable and do not require any preprocessing or run-time lookup that would be required by more efficient message formats. The success of OSC suggests that users are happy with the speed and generally are not interested in greater efficiency at the cost of more complexity.

Clock synchronization techniques are widely known. Madgwick *et al.* [5] describe one for OSC that uses broadcast from a master and assumes bounds on clock drift rates. Brandt and Dannenberg describe a round-trip method with proportional-integral controller [6]. OSC itself supports timestamps, but only in message bundles, and there is no built-in clock synchronization.

Discovery in O2 automatically shares IP addresses and port numbers to establish connections between processes. The liboscqs[1] and OSCgroups[2] library and osctools[3] project support discovery through zeroconf [3] and other systems. Also, Eales and Foss explored discovery protocols in connection with OSC for audio control [4], however their emphasis is on querying the structure of an OSC address space rather than discovery of servers on the network.

Software developers have also discussed and implemented OSC over TCP for reliable delivery. Systems such as liblo[4] offer either UDP or TCP, but not both unless multiple servers are set up, one for each protocol.

4. DESIGN DETAILS

In designing O2, we considered that networking, embedded computers, laptops, and mobile devices have all advanced considerably since the origins of OSC. In particular, embedded computers running Linux or otherwise supporting TCP/IP are now small and inexpensive, and the Internet of Things (IOT) will spur further development of low-cost, low-power, networked sensors and controllers. While OSC deliberately avoided dependency on a particular transport technology to enable low-cost, lightweight communication, O2 assumes that TCP/IP is available to (most) hosts. O2 uses that assumption to offer new features. We also use floating point for simple clock synchronization calculations because floating point hardware has become commonplace even on low-cost microcontrollers, or at least microcontrollers are fast enough to emulate floating point as needed.

4.1 Addresses in O2

In OSC, most applications require users to manually set up connections by entering IP and port numbers. In contrast, O2 provides "services." An O2 *service* is just a unique name used to route messages within a distributed application. O2 addresses begin with the service name, making services the top-level node of a global address space. Thus, while OSC might direct a message to `"/filter/cutoff"` at IP 128.2.1.39, port 3, a complete O2 address would be written simply as `"/synth/filter/cutoff"`, where `"synth"` is the service name.

4.2 UDP vs. TCP for Message Delivery

The two main protocols for delivering data over IP are TCP and UDP. TCP is "reliable" in that messages are retransmitted until they are successfully received, and subsequent messages are queued to insure in-order delivery. UDP messages are often more appropriate for real-time sensor data because new data can be delivered out of

[1] http://liboscqs.sourceforge.net
[2] http://www.rossbencina.com/code/oscgroups
[3] https://sourceforge.net/projects/osctools
[4] http://liblo.sourceforge.net/

order rather than waiting for delivery or even retransmission of older data. O2 supports both protocols.

4.3 Time Stamps and Synchronization

O2 protocols include clock synchronization and time-stamped messages. Unlike OSC, *every* message is time-stamped, but one can always send 0.0 to mean "as soon as possible." Synchronization is initiated by clients, which communicate independently with the master.

5. IMPLEMENTATION

The O2 implementation is small and leverages existing functionality in TCP/IP. In this section, we describe the implementation of the important new features of O2.

5.1 Service Discovery

To send a message, an O2 client must map the service name from the address (or address pattern) to an IP address and port number. We considered existing discovery protocols such as ZeroConf (also known as Rendezvous and Avahi), but decided a simpler protocol based on UDP broadcast messages would be smaller, more portable to small systems, and give more flexibility if new requirements arise.

The O2 discovery protocol uses 5 fixed "discovery" port numbers. We use 5 because we cannot guarantee any one port is unallocated and multiple O2 applications (up to 5) might run on a single host, each requiring a port. When O2 is initialized, O2 allocates a server port and broadcasts the server port, host IP address, local service names and an application name to the 5 discovery ports. Any process running an instance of O2 with the same application name will receive one of these broadcasts, establish TCP and IP sockets connected to the remote process, and store the service name and sockets in a table. Multiple independent applications can share the same local area network without interference if they have different application names. O2 retransmits discovery information periodically since there is no guarantee that all processes receive the first transmissions.

To direct a message to a service, the client simply looks in the lookup table for the appropriate socket and sends the message using TCP or UDP. O2 allows multiple services within a single process without confusion because every message contains its destination service name.

5.2 Timestamps and Clock Synchronization

O2 uses its own protocol to implement clock synchronization. O2 looks for a service named `"_cs"` and when available, sends messages to `"/_cs/ping"` with a reply-to address and sequence number. The service sends the current time and sequence number to the reply-to address. The client then estimates the server's time as the reported time plus half the round-trip time. All times are IEEE standard double-precision floats in units of seconds since the start of the clock sync service. O2 does not require or provide absolute date and time values.

5.3 Replies and Queries

Normally, O2 messages do not send replies, and we do not propose any built-in query system at this time, mainly because queries never caught on in OSC implementations. Unlike classic remote procedure call systems implementing synchronous calls with return values, real-time music systems are generally designed around asynchronous messages to avoid blocking to wait for a reply.

Rather than build in an elaborate query/reply mechanism, we advocate a very simple application-level approach where the "query" sends a *reply-to* address string. The handler for a query sends the reply as an ordinary message to a node under the *reply-to* address. For example, if the *reply-to* address in a `"/synth/cpuload/get"` message is `"/control/synthload"`, then the handler for `"/synth/cpuload/get"` sends the time back to (by convention) `"/control/synthload/get-reply"`. Optionally, an error response could be sent to `"/control/synthload/get-error"`, and other reply addresses or protocols can be easily constructed at the application level.

5.4 Address Pattern Matching and Message Delivery

To facilitate the implementation of O2, we (mostly) adhere to OSC message format. Notice that an O2 server can scan an address string for the "/" after the service name to obtain an OSC-style address pattern. This substring, type information, and data can be passed to many existing OSC implementations for further processing, eliminating the need to implement an all-new message parser. Similarly, existing OSC marshaling code (which converts data to/from messages) can be used to construct messages for O2.

OSC has been criticized for the need to perform potentially expensive parsing and pattern matching to deliver messages. O2 adds a small extension for efficiency: The client can use the form `"!synth/filter/cutoff"`, where the initial "!" means the address has no "wildcards." If the "!" is present, the receiver can treat the entire remainder of the address, "synth/filter/cutoff" as a key and do a hash-table lookup of the handler in a single step. This is merely an option, as a node-by-node pattern match of `"/synth/filter/cutoff"` should return the same handler function.

6. INTEROPERATION

OSC is widely used by existing software. OSC-based software can be integrated with O2 with minimal effort, providing a migration path from OSC to O2. O2 also offers the possibility of connecting over protocols such as Bluetooth[5], MIDI [7], or ZigBee[6].

6.1 Receiving from OSC

To receive incoming OSC messages, call
o2_create_osc_port(`"service"`, *port_num*);
which tells O2 to begin receiving OSC messages on *port_num*, directing them to `service`, which is normally local, but could also be remote. Since O2 uses OSC-compatible types and parameter representations, this adds very little overhead to the implementation. If bundles are present, the OSC NTP-style timestamps must be converted to O2 timestamps before messages are handed off.

[5] http://www.bluetooth.org
[6] http://www.zigbee.org

6.2 Sending to OSC

To forward messages to an OSC server, call
o2_delegate_to_osc(`"service"`, *ip*, *port_num*);
that tells O2 to create a virtual service (name given by the `service` parameter), which converts incoming O2 messages to OSC messages and forwards them to the given *ip* address and *port_num*. Now, any O2 client on the network can discover and send messages to the OSC server.

6.3 Other Transports

Handling OSC messages from other communication technologies poses two interesting problems: What to do about discovery, and what exactly is the protocol? The O2 API can also be supported directly on clients and servers connected by non-IP technologies. As an example, let us assume we want to use O2 on a Bluetooth device (we will call it Process D, see Figure 1) that offers the "Sensor" service. We require a direct Bluetooth connection to Process B running O2. Process B will claim to offer the "Sensor" service and transmit that through the discovery protocol to all other O2 processes connected via TCP/IP. Any message to "Sensor" will be delivered via IP to Process B, which will then forward the message to Host D via Bluetooth. Similarly, programs running on Host D can send O2 messages to Process B via Bluetooth where the messages will either be delivered locally or be forwarded via TCP/IP to their final service destination. It is even possible for the destination to include a final forwarding step though another Bluetooth connection to another computer, for example there could be services running on computers attached to Process C in Figure 1.

Non-IP networks are supported by optional libraries, essentially giving O2 a "plug-in" architecture to ensure both a small core and flexibility to create extensions.

In addition to addressing services, O2 sometimes needs to address the O2 subsystem itself, e.g. clock synchronization runs even in processes with no services. Services starting with digits e.g. `"128.2.60.110:8000"`, are interpreted as an IP:Port pair. To reach an attached non-IP host, a suffix may be attached, e.g. Host D in Figure 1 might be addressed by `"128.2.60.110:8000:bt1"`.

7. CURRENT STATUS

A prototype of O2 in the C programming language is running the discovery algorithm and sending messages. Performance measurements show that CPU time is dominated by UDP packet send and receive time, even when messages are sent to another process on the same host (no network link is involved). We were unable to measure any impact of discovery or service lookup in a test where two processes send a message back and forth as fast as possible. In this test, total message delivery (real or "wall") time is about 13μs, or 77,000 messages per second, on a 2.4 GHz Intel Core i7 processor, which is faster than OSC using liblo due to some minor differences in the way messages are accepted from the network.

We believe O2 is a good candidate for OSC-like applications in the future. A number of extensions are possible, and future work includes extensions to allow discovery beyond local area networks, audio and video streaming, and dealing with network address translation (NAT).

O2 is available: https://github.com/rbdannenberg/o2.

8. SUMMARY AND CONCLUSIONS

O2 is a new protocol for real-time interactive music systems. It can be seen as an extension of Open Sound Control, keeping the proven features and adding solutions to some common problems encountered in OSC systems. In particular, O2 allows applications to address services by name, eliminating the need to manually enter IP addresses and port numbers to form connected components. In addition, O2 offers a standard clock synchronization and time-stamping system that is suitable for local area networks. O2 offers two classes of messages so that "commands" can be delivered reliably and sensor data can be delivered with minimal latency. We have implemented a prototype of O2 that is similar in size, complexity and speed to an Open Sound Control implementation. Although O2 assumes that processes are connected using TCP/IP, we have also described how O2 can be extended over a single hop to computers via Bluetooth, ZigBee or other communication links.

Acknowledgments

Thanks to Adrian Freed for comments on a draft of this paper.

9. REFERENCES

[1] M. Wright, A. Freed and A. Momeni, "OpenSound Control: State of the Art 2003," in *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, Montreal, Canada, 2003, pp. 153-159.

[2] M. Henning, "The rise and fall of CORBA," *ACM Queue,* vol. 4, no. 5, 2006, pp. 29-34.

[3] E. Guttman, "Autoconfiguration for IP Networking: Enabling Local Communication," *IEEE Internet Computing,* vol. 5, no. 3, 2001, pp. 81–86

[4] A. Eales and R. Foss, "Service discovery using Open Sound Control," *AES 133rd Convention,* San Francisco, 2012.

[5] S. Madgwick, T. Mitchell, C. Barreto, and A. Freed, "Simple synchronisation for open sound control. *41st International Computer Music Conference 2015*, Denton, Texas, 2015, pp . 218-225.

[6] E. Brandt and R. Dannenberg, "Time in Distributed Real-Time Systems," Proceedings of the International Computer Music Conference , 1999.

[7] J. Rothstein, *MIDI: A Comprehensive Introduction, 2nd ed.*, A-R Editions, 1995.

Introducing D^4: An Interactive 3D Audio Rapid Prototyping and Transportable Rendering Environment Using High Density Loudspeaker Arrays

Ivica Ico Bukvic
Virginia Tech
SOPA, DISIS, ICAT
ico@vt.edu

ABSTRACT

With a growing number of multimedia venues and research spaces equipped with High Density Loudspeaker Arrays, there is a need for an integrative 3D audio spatialization system that offers both a scalable spatialization algorithm and a battery of supporting rapid prototyping tools for time-based editing, rendering, and interactive low-latency manipulation. D^4 library aims to assist this newfound whitespace by introducing a Layer Based Amplitude Panning algorithm and a collection of rapid prototyping tools for the 3D time-based audio spatialization and data sonification. The ensuing ecosystem is designed to be transportable and scalable. It supports a broad array of configurations, from monophonic to as many as hardware can handle. D^4's rapid prototyping tools leverage oculocentric strategies to importing and spatially rendering multidimensional data and offer an array of new approaches to time-based spatial parameter manipulation and representation. The following paper presents unique affordances of D^4's rapid prototyping tools.

1. INTRODUCTION

The history of Western music can be seen as a series of milestones by which the human society has emancipated various dimensions of aural perception. Starting with pitch and rhythm as fundamental dimensions, and moving onto their derivatives, such as homophony, and polyphony, each component was refined until its level of importance matched that of other already emancipated dimensions. In this paper the author posits that the observed maturity or the emancipation of these dimensions is reflected in their ability to carry structural importance within a musical composition. For instance, a pitch manipulation could becom a motive, or a phrase that is further developed and varied and whose permutations can independently drive the structural development. The same structural importance can be also translated into research contexts where a significant component of the data sonification if not its entirety can be conveyed within the emancipated dimension. With the aforesaid definition in mind, even though timbre plays an important role in the development of the Western music, particularly the orchestra, its steady use as a structural element does not occur until the 20th century. Indeed, the 20th century can be seen as the emancipation of timbre. Similarly, while the audio spatialization has played a role throughout the history of music, with occassional spikes in its importance, including the Venetian cori spezzati [1] or the spatial interplay among the orchestral choirs, its structural utilization is a relatively recent phenomena. Today, the last remaining dimension of the human aural perception yet to undergo its emancipation is spatialization. From augmented (AR) and virtual reality (VR), and other on- and in-ear implementations, to a growing number of venues supporting High Density Loudspeaker Arrays (HDLAs), 21st century is poised to bring the same kind of emancipation to the spatialization as the 20th century did to timbre. Similarly, data audification and sonification using primarily spatial dimension are relatively new but nonetheless thriving research areas whose full potential is yet to be realized [2].

In this paper HDLAs are defined as loudspeaker configurations of 24+ loudspeakers capable of rendering 3D sound without having to rely solely on virtual sources or post-processing techniques. This definition suggests there are multiple layers of loudspeakers spread around the listening area's perimeter.

Apart from the ubiquitous amplitude panning [3], contemporary audio spatialization algorithms include Ambisonics [4], Head Related Transfer Function (HRTF) [5], Vector Based Amplitude Panning (VBAP) [6], Depth Based Amplitude Panning (DBAP) [7], Manifold-Interface Amplitude Panning (MIAP) [8], and Wave Field Synthesis (WFS) [9].

There is a growing number of tools that leverage the aforesaid algorithms. This is of particular interest because the lack of such tools makes it particularly cumbersome to integrate algorithms in the well-established research and artistic production pipelines. The most common implementations are found in programming languages like Max [10] and Pure-Data [11] where they offer spatialization capabilities (e.g. azimuth and elevation), leaving it up to user to provide more advanced time-based editing and playback. Others focus on plugins for digital audio workstations (DAWs) (e.g. [12, 13]) thus leveraging the environment's automation, or offer self-standing applications dedicated to audio editing and rendering, such as Sound Particles [14], Meyer's Cuestation [15], Zirkonium [16], and Sound Emotion's Wave 1 [17]. The fact that a majority of these tools have been developed in the past decade points to a rapidly developing field. A review of the existing tools has uncovered a whitespace [18], a unique set of desirable features an algorithm coupled with time-editing tools ought to deliver in order to foster a more widespread adoption and with it standardization:

- The support for irregular High Density Loudspeaker Arrays;
- Focus on the ground truth with minimal amount of idiosyncrasies;
- Leveraging the vantage point to promote data comprehension;
- Optimized, lean, scalable, and accessible, and
- Ease of use and integration through supporting rapid-prototyping time-based tools.

2. D^4

D^4 is a new Max [10] spatialization library that aims to address the aforesaid whitespace by:

1. Introducing a new lean, transportable, and scalable Layer Based Amplitude Panning (LBAP) audio spatialization algorithm capable of scaling from monophonic to HLDA environments, with particular focus on advanced perimeter-based spatial manipulations of sound that may prove particularly useful in artistic, as well as audification and sonification scenarios, and
2. Providing a collection of supporting rapid prototyping time-based tools that leverage the newfound audio spatialization algorithm and enable users to efficiently design and deploy complex spatial audio images.

D^4's Layer Based Amplitude Panning (LBAP) algorithm groups speakers according to their horizontal layer and calculates point sources using the following series of equations applied to the four nearest speakers:

Below layer:

$$BL_{amp} = cos(BL_{distance} * \pi/2) * cos(B_{amp} * \pi/2) \quad (1)$$

$$BR_{amp} = sin(BL_{distance} * \pi/2) * cos(B_{amp} * \pi/2) \quad (2)$$

Above layer:

$$AL_{amp} = cos(AL_{distance} * \pi/2) * cos(A_{amp} * \pi/2) \quad (3)$$

$$AR_{amp} = sin(AL_{distance} * \pi/2) * cos(A_{amp} * \pi/2) \quad (4)$$

In the aforesaid equations B stands for the nearest layer below the point source's elevation and A the nearest layer above. BL stands for the nearest left speaker on the below layer, BR for nearest right speaker on the below layer, AL for the nearest left speaker on the above layer, and AR for the nearest right on the above layer. Amp refers to the amplitude expressed as a decimal value between 0 and 1. Distance reflects the normalized distance between two neighboring speakers within the same layer expressed as a decimal value between 0 and 1.

LBAP focuses on the use of a minimal number of speakers. For point sources it can use anywhere between one to four speakers. Arguably its greatest strength resides in its ability to accommodate just about any speaker configuration (from monophonic to as many loudspeakers as the hardware can handle) for perimeter-based spatialization with minimal CPU overhead. Like most other algorithms, its positioning is driven primarily by the azimuth and elevation values, with the ear level being 0° elevation and 0° azimuth being arbitrarily assigned in respect to venue's preferred speaker orientation.The algorithm is further described in greater detail in [18].

Similar to VBAP's Source Spread, LBAP also offers Radius option that accurately calculates per-speaker amplitude based on spherical distance from the point source. The Radius distance is expressed in spherical degrees from the center of the point source and loudspeaker position. It also introduces a unique feature called Spatial Mask (discussed below). When coupled with the D^4 library, LBAP is further enhanced by a series of unique affordances, including Motion Blur (also discussed below) and a battery of time-based editors that leverage oculocentric user interfaces for generating, importing, and manipulating multidimensional data.

D^4 library focuses on mostly open source (MOSS) lean implementation that leverages maximum possible amount of built-in Max objects while introducing only two new java-based objects, namely the main spatialization object D4 and Jitter-based mask editor D4_med_matrix. This design choice introduces new challenges, like the lack of graceful handling of determinacy within the Max's multithreaded environment (e.g. using a poly object for dynamic instantiation of per-speaker mask calculating abstractions). It also provides opportunities for the user to build upon and expand library's functionality, thus minimizing the limitations typically associated with closed (a.k.a. blackbox) alternatives.

Other features aimed at addressing the aforesaid whitespace include the support for a broad array of speaker configurations, dynamic runtime reconfigurability of the speaker setup, user-editable loudspeaker configuration syntax, focus on perimeter-based spatialization without the need for a special spectral adjustment or per-loudspeaker processing beyond amplitude manipulation, low-latency real-time-friendly operation (in tests the system was able to render stable audio output with 11ms latency at 48KHz 24bit sampling rate using 128 speakers), built-in audio bus system per each audio source designed to promote signal isolation and streamline editing, independent layers (e.g. sub arrays), and focus on leveraging real-world acoustic conditions where vantage point is treated as an asset rather than a hindrance. LBAP does not aim to compensate for vantage point perceptual variances. This is in part because such an implementation mimics real-world acoustic conditions, and is therefore seen as offering opportunities for broadening of cognitive bandwidth by cross-pollinating different modalities (e.g. location-based awareness and aural perception), and also in part because it minimizes the need for idiosyncrasies that may limit system's scalability and transportability, and/or adversely affect its overall CPU overhead. D^4's lean design promotes optimization and scalability, as well as easy expansion, with the ultimate goal of promoting transportability. The library can serve as a drop-in replacement for the mainstream spatial-

ization alternatives that rely on azimuth and elevation parameters. Furthermore, D^4 tools promote ways of retaining time-based spatial configuration in its original, editable format that can be used for real-time manipulation. The same can be also used to render time-based data for different speaker configurations and later playback that bypasses potentially CPU intensive real-time calculation. To aid in this process the system offers tools for playback of prerendered spatial data thereby making its playback resolution limited only by the per-loudspeaker amplitude crossfade values whose primary purpose is to prevent clicks while also enabling novel features like the Motion Blur.

3. UNIQUE AFFORDANCES

3.1 Spatial Mask

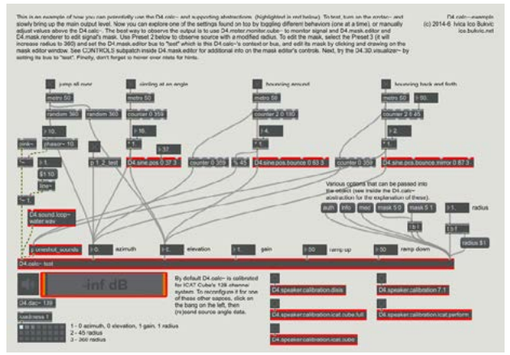
Spatial Mask (SM) is one of the unique features of the D^4 ecosystem. Akin to that of its visual counterpart LBAP considers the entire perimeter space to have the default mask of 1. This means wherever the point source is and whatever its radius, it will populate as many loudspeakers as its computed amplitude and radius permit based solely on its calculated amplitude curve. The spatial mask, however, can be changed with its default resolution down to 0.5° horizontally and 1° vertically, giving each loudspeaker a unique maximum possible amplitude as a float point value between 0 and 1. As a result, a moving source's amplitude will be limited by loudspeaker's corresponding mask value as it traverses the said loudspeaker. This also allows a situation where a point source with 180° radius that emanates throughout all the loudspeakers can now be dynamically modified to map to any SM, thus creating complex shapes that go well beyond the traditional spherical sources.

3.2 Time-Based Editing Tools

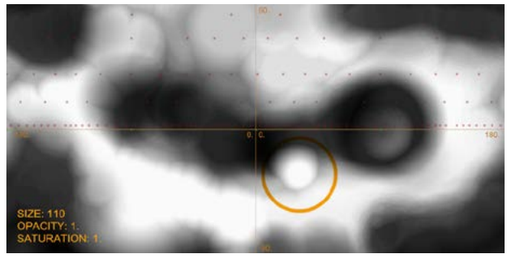
SM implementation leverages Jitter library and its affordances, making it convenient to import and export SM snapshots and automate time-based alterations. Like a single channel video, D^4's SM editing tools use grayscale 2D matrix to calculate the ensuing per-loudspeaker mask. As of version 2.1.0, the time-based editing tools allow for SM translation and can couple azimuth, elevation, as well as up-ramp and down-ramp data into a single coll-formatted file that is accompanied by matrices corresponding with each keyframe. The library can then interpolate between those states at user-specified resolution both in real-time and via batch rendering, allowing for time-stretching and syncing with content of varying duration.

The entire D^4 ecosystem is virtual audio bus aware and widgets (where appropriate) can be easily reconfigured to monitor and/or modify properties of a specific bus. Where applicable, leaving the bus name blank will revert to monitoring main outs. Apart from the D4.calc abstraction that encompasses library's core functionality and instantiates a single movable source and a bus, the main supporting tools include (Figure 1):

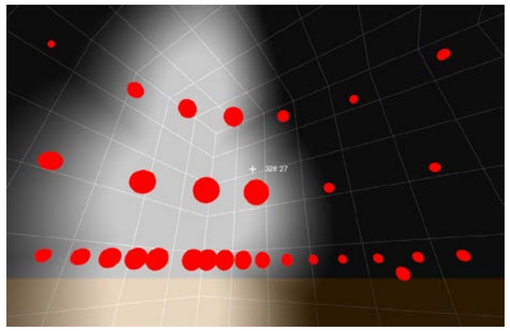
- **D4.mask.editor** (a.k.a. the editor) designed to provide a basic toolset for visual mask editing. It leverages Jitter library to provide SM painting ability and link it with a particular bus or a sound source;



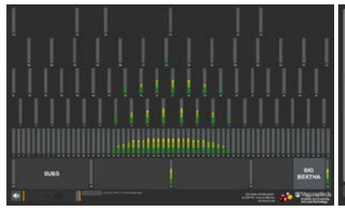
(a) D4.calc -example



(b) D4.mask.editor



(c) D4.3D.visualizer



(d) VT ICAT Cube monitor

(e) 7.1 monitor

(f) D4.mask.renderer

(g) D4.mask.player

Figure 1: A collection of D^4's widgets.

- **D4.3D.visualizer** that allows users to monitor both SM and the bus amplitude output in a spatially aware 3D environment;
- **D4.meter.monitor.*** is a collection of abstractions that offer a more traditional way of monitoring levels. They are built using D^4's helper abstractions designed to promote rapid-prototyping of configurations other than the ones already included with the library. As of version 2.1.0, the library offers visualizer for three Virginia Tech signature spaces and a prototype for 7.1 surround sound system;
- **D4.speaker.calibration.*** provide calibration settings for a growing number of venues, as well as more common multichannel configurations (e.g. 7.1);
- **D4.mask.renderer** (a.k.a. renderer) is the nexus for all time-based editing and rendering. All of the aforesaid tools, including the D4.calc abstraction are designed to interface with this object, feed edited data and update their own state based on the data provided by the renderer, and
- **D4.mask.player** that can play data rendered by the D4.mask.renderer and feed it into the target D4.calc (a.k.a. bus).

The entire library is envisioned as a modular collection of self-standing, yet mutually aware widgets. User can customize their workspace as they deem fit. The same widgets can be also embedded as GUI-less abstractions in their own patches by leveraging the annotated inlets and outlets, as well as included documentation and examples. In addition, due to their MOSS design the widgets themselves can be further enhanced (e.g. by altering the default speaker configuration that is preloaded within each D4.calc , adding custom filters to specific outputs, or by introducing new and more advanced ways of processing SM matrices). The resulting community enhancements that prove particularly useful may be eventually merged into the future upstream releases.

The ability of each widget to be utilized independently from others is limited only by context. For instance, editing SM makes no sense unless the bus being edited actually exists. Likewise, storing SM is impossible without having a renderer monitoring the same bus. To maximize the possible number of viable configurations has required some widgets to carry redundant implementations. For instance, the editor if used solely to alter Mask on a particular bus without the intent to store it (e.g. for a real-time manipulation), requires D4.mask.calculator abstraction that is also present within the renderer. Consequently, to minimize the redundancy and the ensuing CPU overhead in situations where both abstractions are present within the same bus pipeline, the library has a framework to autodetect such a condition and minimize the redundancy by disabling the calculator within the editor and forwarding the editor data directly to the renderer.

3.3 Helper Abstractions

In addition to the aforesaid widgets, D^4 also offers a collection of helper abstractions designed to streamline library's utilization in more complex scenarios. D4 , D4.dac , and D4.cell are abstractions used for the dynamic creation of the bus outputs, as well as main outputs, both of whose state can be monitored and manipulated (e.g. bus-specific up- and down-ramps that can be used to manipulate a moving source attack and trail envelope, effectively resulting in the aural equivalent of the Motion Blur [18]). D4.meter.cell is designed to be used primarily as a Max's visual abstraction (a.k.a. bpatcher) for the purpose of rapid prototyping spatially-aware visual level monitors, whereas D4.meter.3D.cell is used for monitoring levels and forwarding those to the D4.mask.3D.visualizer. The library also includes a series of javascripts for managing dynamic generation of necessary buses and outputs. Other smaller convenience abstractions include oneshot audio events (D4.sound.oneshot) and audio loops (D4.sound.loop).

D4.sine.pos* collection of abstractions provide a more advanced automated spatialized source motion. As shown in the introductory D4.calc -example patch, when connected to the D4.calc 's azimuth and elevation values, these abstractions can provide circular perimeter-based motion at an angle other than the traditional horizontal trajectory. *bounce version mimics a bouncing object, while *mirror variant enables bouncing against both ends of the desired elevation range. Both abstractions can take optional arguments that modulate the range and offset.

4. CONCLUSIONS AND FUTURE WORK

D^4 is an actively maintained production ready Max library designed to address the limited transportability of spatial audio using HDLAs in artistic and reseach contexts. It does so by coupling a new Layer Based Amplitude Panning algorithm with a battery of supporting time-based tools for importing, editing, exporting, and rendering spatial data, including real-time low-latency HDLA scenarios. The newfound affordances, such as the Radius, Spatial Mask, and Motion Blur, when combined with Jitter-based editing tools, offers opportunities for exploring new approaches to audio spatialization. These include scientific research that furthers the understanding of human spatial perception and more importantly leveraging the ensuing knowledge for the purpose of emancipating spatial audio dimension both within the artistic and research scenarios while providing a scalable and transportable way of disseminating HDLA content.

Given D^4's expanding feature set, it is unclear whether the current MOSS approach as a Max library will prove an environment conducive of creativity it aims to promote, particularly in respect to a battery of tools and widgets that in their current form defy more traditional approaches to user interfaces commonly associated with DAWs and other time-based editing tools. Based primarily on user demand, it is author's intention to continue investigating optimal ways of introducing timeline-centric features within the existing implementation and expanding to other frameworks, including potentially a self-standing application.

5. OBTAINING D^4

D^4 can be obtained from `http://ico.bukvic.net/main/d4/`.

6. REFERENCES

[1] D. Bryant, "The ?cori spezzati? of St Mark's: myth and reality," *Early Music History*, vol. 1, pp. 165–186, Oct. 1981. [Online]. Available: http://journals.cambridge.org/article_S0261127900000280

[2] G. Kramer, *Auditory display: Sonification, audification, and auditory interfaces*. Perseus Publishing, 1993. [Online]. Available: http://dl.acm.org/citation.cfm?id=529229

[3] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, 1997. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=7853

[4] N. Barrett, "Ambisonics and acousmatic space: a composers framework for investigating spatial ontology," in *Proceedings of the Sixth Electroacoustic Music Studies Network Conference*, 2010. [Online]. Available: http://www.natashabarrett.org/EMS_Barrett2010.pdf

[5] B. Carty and V. Lazzarini, "Binaural HRTF based spatialisation: New approaches and implementation," in *DAFx 09 proceedings of the 12th International Conference on Digital Audio Effects, Politecnico di Milano, Como Campus, Sept. 1-4, Como, Italy*. Dept. of Electronic Engineering, Queen Mary Univ. of London,, 2009, pp. 1–6. [Online]. Available: http://eprints.maynoothuniversity.ie/2334

[6] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, 1997. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=7853

[7] T. Lossius, P. Baltazar, and T. de la Hogue, "DBAPdistance-based amplitude panning." Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2009. [Online]. Available: http://www.trondlossius.no/system/fileattachments/30/original/icmc2009-dbap-rev1.pdf

[8] Z. Seldess, "MIAP: Manifold-Interface Amplitude Panning in Max/MSP and Pure Data," in *Audio Engineering Society Convention 137*. Audio Engineering Society, 2014. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?conv=137&papernum=9112

[9] K. Brandenburg, S. Brix, and T. Sporer, "Wave field synthesis," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2009*. IEEE, 2009, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5069680

[10] M. Puckette, "Max at seventeen," *Computer Music Journal*, vol. 26, no. 4, pp. 31–43, 2002. [Online]. Available: http://www.mitpressjournals.org/doi/pdf/10.1162/014892602320991356

[11] ——, "Pure Data: another integrated computer music environment," *IN PROCEEDINGS, INTERNATIONAL COMPUTER MUSIC CONFERENCE*, pp. 37–41, 1996. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.3903

[12] M. Kronlachner, "Ambisonics plug-in suite for production and performance usage," in *Linux Audio Conference*. Citeseer, 2013, pp. 49–54. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.654.9238&rep=rep1&type=pdf#page=61

[13] J. C. Schacher, "Seven years of ICST Ambisonics tools for maxmsp?a brief report," in *Proc. of the 2nd International Symposium on Ambisonics and Spherical Acoustics*, 2010. [Online]. Available: http://ambisonics10.ircam.fr/drupal/files/proceedings/poster/P1_7.pdf

[14] "Sound Particles - Home," http://www.sound-particles.com/.

[15] "D-Mitri : Digital Audio Platform | Meyer Sound," http://www.meyersound.com/product/d-mitri/spacemap.htm. [Online]. Available: http://www.meyersound.com/product/d-mitri/spacemap.htm

[16] C. Ramakrishnan, "Zirkonium: Non-invasive software for sound spatialisation," *Organised Sound*, vol. 14, no. 03, pp. 268–276, Dec. 2009. [Online]. Available: http://journals.cambridge.org/article_S1355771809990082

[17] E. Corteel, A. Damien, and C. Ihssen, "Spatial sound reinforcement using Wave Field Synthesis. A case study at the Institut du Monde Arabe," in *27th TonmeisterTagung-VDT International Convention*, 2012. [Online]. Available: http://www.wfs-sound.com/wp-content/uploads/2015/03/TMT2012_CorteelEtAl_IMA_121124.pdf

[18] I. I. Bukvic, "3D TIME-BASED AURAL DATA REPRESENTATION USING D 4 LIBRARY?S LAYER BASED AMPLITUDE PANNING ALGORITHM," in *the 22nd International Conference on Auditory Display (ICAD 2016). July 3-7, 2016, Canberra, Australia*, 2016. [Online]. Available: http://www.icad.org/icad2016/proceedings2/papers/ICAD2016_paper_10.pdf

Improvements of iSuperColliderKit and its Applications

Akinori Ito
Tokyo University of Technology
akinori@edu.teu.ac.jp

Kengo Watanabe
Watanabe-DENKI Inc.
kengo@wdkk.co.jp

Genki Kuroda
Tokyo University of Technology
g3115002e8@edu.teu.ac.jp

Ken'ichiro Ito
Tokyo University of Technology
itoken@stf.teu.ac.jp

ABSTRACT

iSuperColliderKit (abbr. iSCKit) has been improved in the aspect of productivity and maintainability. In this version, we implemented 3 features; smart initialization without declaring as a shared instance, file reading, avoiding necessity to handle pointers in objective-C. The features become easier to embed due to re-organizing the project template and build settings.

1. CONCEPT

1.1 The Original Motivation of This Project

There have already been some sound API or game sound middleware[1][2] have the features of modifying the sampled data: changing tempo, transpose per sound file, dynamic filtering and mixing. However, it is still difficult to make some musical variations like "virtual improvisator" on iOS. If programmers would develop some kind of applications, they have to develop their own algorithmic composition features and combine the low-level MIDI API. In the meantime, numerous platforms have been proposed to bring synthesis to iOS. libPd[3], AudioKit[4] and MoMu[5] are widely used in iOS developer community. However, they are suitable for building synthesizer or effector, not for playing multiple musical series and changing musical element dynamically. urMus is full fledged meta-programming environment[6]. That use OpenGLES as a graphic API. Gibber[7] and J.Allison's web applications[8] are some approaches the problem through the web. These 3 research have great portability but our approach is aim to integrate the native iOS graphic API such as SpriteKit and SceneKit.
The main motivation of this project is to be enable to embed the generative music function of SuperCollider in native iOS applications as a sound server. In the aspect of computer music, the designing UI or graphical elements for generative music on iOS become freer. In the iOS developers' side, they can use the dynamically changing musical elements on any types of applications: games, art pieces, even some utility software. The target developers for iSuperColliderKit (abbr. iSCKit) have 2 specialties, one is as an iOS developer, the other is as an experienced person of SuperCollider. The main target applications have the feature of multi-touch interaction. Therefore, the type is able to define as an "interactive music applications". A kind of clear instance is some parts of game. However, we do not limit the target for game. There are several instance, education, interactive installation, dynamic storytelling and so on. In the actual applications, it needs many kind of underscores, atmospheric sound. In the case of using the network features in SuperCollider, programmers are able to apply to data driven dynamic music applications on users' palmtop.

1.2 Problem of previous work

Based on the concept and target, we run the development[9] and take it public[10]. At that time, iSCKit was still unstable and inconvenience to use for other developers. The engine had to be instantiated as a shared instance and developers had to manage handle pointer. Further, there was no SuperCollider file handling function. The OSC code increased in length. It caused to some difficulties to manage the long and multiple layered music descriptions. In this article, we report how to improve iSCKit.

2. SUMMERY OF PREVIOUS WORK

To clarify the points of the improvements, we summarize our previous work.

2.1 Replacement the 32bit ARM NEON code

At the beginning of this trial, there were many of codes 32bit ARM NEON architecture in the past version that we referred on GitHub repository[11], especially in `SC_VFP11.h`, `IOUGen.cpp`, `SC_CoreAudio.cpp`. The "`vIn_next_a()`, `vfill()`, `vcopy()`" are the functions for 32bit version NEON architecture of standard "`In_next_a()`, `Fill()`, `Copy()`". However, these functions caused many of errors for the latest build environment. Therefore we replaced many of `vIn_next_a()`, `vfill()`, `vcopy()` to the standard functions.

2.2 Adapting to ARC Programming Style

In the iOS programming, the memory management mechanism "Automatic Reference Counter" was supported from Xcode4.2. In accordance with it, the "AutoReleasePool" used before became *deprecated*. Therefore, we deleted 51 `autorelease`, 47 `retain`, 116 `release`, and many of corresponding `dealloc`.

2.3 Separating SC Server from Editor

The previous version that we referred seemed to construct as a "perfect clone" of SuperCollider on iOS because of including any UI parts of PC version: Live Coding Editor, Control Panel with boot, exec button etc.

Figure 1. UI of Previous version.

Each UI parts were constructed by Interface Builder and SCController class is deeply connected the mechanism of Interface Builder.

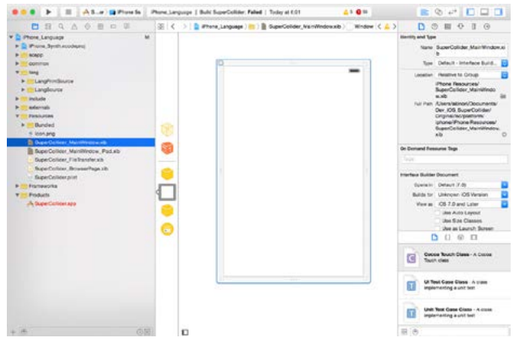


Figure 2. Each UI parts built with Interface Builder in Previous version.

This architecture is suitable for authentic SuperCollider users who want to do live coding on iOS environment as the same of PC. However, our goal aim toward to another direction. Therefore, we separated UI elements from SC server to make it as a sound engine.

2.4 Miscellaneous adaptations

During this project, the compiler environment changed drastically. Xcode adopted LLVM clang instead of GCC. The architecture of iOS devices moved to 64bit environment. In association with that, we have done many of casting all of the project for 64bit and included the latest libsndfile from Csound repository. As a result, iSCKit enabled to make the SuperCollider server for iOS in just 3 shared libraries: `libiSCKit.a`, `libscsynth.a` and `libsndfile.a`.

2.5 Review

At that time, we succeeded to construct the build environment for iOS7 and later, sending some SuperCollider code fragments as `NSString` from the interaction method delegating `UIView` instance by objective-C or Swift 1.2 codes. However, it was just technical tests. The architecture was not enough sophisticated. The programmers had to take care for the management the `sharedInstance` through the building their applications and do the complicated build settings manually.

3. IMPROVEMENTS

Our team continued the improvement some problems as described above from February 2015 after submitting our previous works.

3.1 iSC class

On the previous version of our work, the controller as MVC model was `iSCController` class. In the initializing phase of an iOS application, a programmer had to initialize the iSCController instance. This is an example of initialization code by objective-C.

```
#import "iSCAppDelegate.h"

@implementation iSCAppDelegate
@synthesize window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];

    [iSCController *scc = [iSCController sharedInstance];];
    [scc interpret:@"s.boot"];
```

In this version, an iOS programmers had to prepare the instance of controller class as a shared instance because of ensuring the access from all of the projects. This situation is a typical Singleton pattern. Actually, it is assumed that one instance of `iSCController` is enough to one iOS application like a sound driver control class. A programmer had to get the pointer of this instance in initializing phase. In the aspect of education, this specification is good for notification that `iSCController` class returns its pointer of instance but not suitable for usual programming. Further, when the programmer uses this instance, interpret method needed a NSString data type as below.

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    NSString* message =
            [NSString stringWithFormat:@"{SinOsc.ar(
                    440*%d, 0, EnvGen.ar(Env.new([0, 1, 0], [0.01, 0.5]), doneAction:2)}}.play; %d;];
    [scc interpret:message];
```

According the design of this system, sending messages always be text massage. Hence, we made `iSC` class instead of `iSCController` class prepared some class methods for setup as a Singleton, easy initializing and sending SuperCollider code fragments all of the time: `setup()`, `interpretC()` and `interpret()`. As a result, an initializing code become shorten and easy understanding. The below examples are the initializing code by previous `iSCController` class.

```
[iSCController *scc =
       [iSCController sharedInstance];]; (Obj-C)

let scc = iSCController.sharedInstance()
scc.setup()                                (Swift)
```

The codes of this process used by `iSC` class becomes simple as below.

```
[iSC setup];       (Obj-C)
iSC.setup()        (Swift)
```


Further, we add the scd file reading feature `interpretFile()`. SuperCollider can work by relatively less amount of codes. This feature fits to embed some short code fragments in iOS interaction mechanism. However, the initial SynthDef, whole music data or any data table (collection, list, array etc.) often become several dozens line. In this case, especially initializing phase, this file reading function is useful. The source code in objective-C is below.

```
+ (void) interpretFile:(NSString*)path_arg{
    NSError *error;
    NSString *file_str=[NSString stringWithContentsOfFile:path_arg
                                                 encoding:NSUTF8StringEncoding
                                                    error:&error];
    if(error != nil) { NSLog(@"%@", error.description); return; }
    [internal_sc_controller interpretSCMessage:file_str];}
```

The usage of `interpretFile()` in Swift is below.

```
iSC.interpretFile("mainTrack.scd")
```

The features of new `iSC` class instead of `iSCController` class are summarized as follow.

- Easier initialization
- Easier file reading
- Avoiding necessity to handle pointer (objective-C)

3.2 Project Template

In the previous version, the project hierarchy was not organized and programmers had to set up the complicated Xcode build settings and manage the file/folder placement. To become easier the original application development, we re-organized the folder hierarchy and made some templates for embedding the iSCKit features in their own projects.

3.2.1 Project hierarchy

The `iSuperColliderKit-master` folder is the top directory of uncompressed archive or git clone. There are several sub-folders but the just 2 folders are checked by programmers: `projects` and `lib`. The latter folder is automatically produced on building the shared libraries. There are 3 sub-folders in projects: `iSCKit`, `iSCApp`, `iSCAppSwift`. iSCKit folder contains the project for building the SuperCollider server for iOS. `iSCApp` contains the template project build by objective-C. `iSCAppSwift` is also the project template build by Swift.

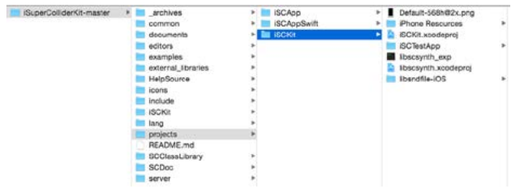


Figure 3. The default hierarchy of iSCKit when programmers just uncompress the zip archive or git clone. The `projects` folder is the 'key' folder of this project.

3.2.2 Building 3 shared libraries

First, launch the `iSCKit.xcodeproj` in the `projects` folder to build the 3 shared libraries. The special settings is not needed for producing them. The project automatically makes `lib` folder on the same level of `projects` folder.

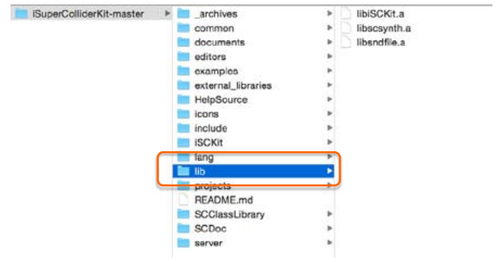


Figure 4. The iSCKit project makes `lib` folder and 3 shared libraries on it.

This project builds for an actual device only. Do not select any simulators.

3.2.3 Application templates

The iSCApp and iSCAppSwift includes the complicated build settings: build path, build options and so on. It is easy to create their own applications that programmers make a copy of project and work at projects folder. If they would place the copies of iSCApp or iSCAppSwift, they can customize their own environments to modify 2 parameters and 1 additional operation. One is the setting of Library Search Path. The project templates refer the directory of "`../../lib`" by default. It means "`iSuperColliderKit-mastar/lib/`". They can place 3 library files anywhere by assign them manually.

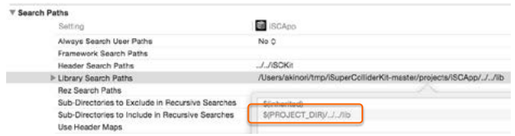


Figure 5. The default Library Search Path on the project templates.

The another is the setting of Header Search Path. The templates indicate it as "`iSuperColliderKit-master/`". Originally programmers have to include or import several headers on their own programs manually. To avoid missing including and importing, we prepare the `iSCKit.h` on `iSCKit` folder and assign the Header Search Path on the top directory of this project. The notation is just `#import <iSCKit/iSCKit.h>`. Therefore, the original project set up is below

- Adding Library Search Path: `$(anywhere)/iSuperColliderKit-master/lib/`
- Adding Hearder Search Path: `$(anywhere)/iSuperColliderKit-master/`
- Writing this code: `#import <iSCKit/iSCKit.h>`

4. APPLICATIONS

To explore the new experiment of this purpose, we developed some test applications. They control the modal transition, rhythmic chance variables, reaction of collision detection and so on.

4.1 Changing a drum-beat density with rotation

This is a test rhythmic modulation with multi touch rotation. To detect the rotation by 2 fingers, iOS programmers can use `UIRotationGestureRecognizer` API with UIKit. The rotation method of its instance returns the angle. To keep former rotation value, it can simulate the KNOB style UI. In this application, drum pattern is generated by a probability table in a SuperCollider document. The probability is selected by variables `~tension`. The iOS application calculates the tension by user rotation interaction. As a result, the drum patterns are changed in real time by user multi-touch rotation gesture. The effect is like a "drummer" in LogicProX

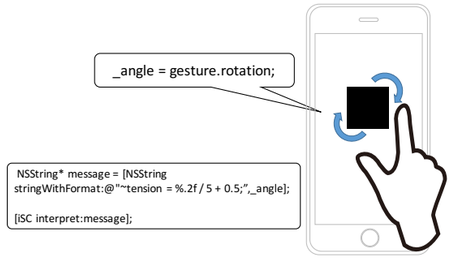


Figure 6. Drum pattern changer by rotate interaction.

4.2 Pinball with modal change

It is a test case of modal change and reaction of collision detection like a pinball game. At the launch the application, it reads the "mainTrack" as a BGM. This SuperCollider document contains the variables `scl` which decides the musical mode. On the other hand, the iOS application has the collision detection feature in hand. When a collision is detected, the embedded `iSC.interpret()` method send to SC server to change the musical mode. As a result, the mode of whol BGM changes dynamically. This function run on applications using SpriteKit and SceneKit API.

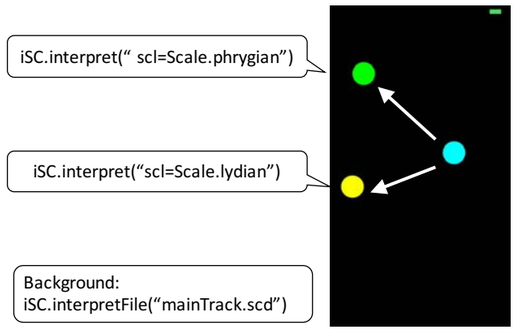


Figure 7. A modal change example like a pinball game.

5. CONCLUSION AND FUTURE WORK

iSCKit has been improved in the aspect of productivity and maintainability. The latest version of iSCKit corresponds Xcode 7, Swift 2.0 and iOS9. The features become easier to embed due to re-organizing the project template and build settings. Due to the new controller class `iSC` which is assumed to be use as a Singleton, it become easier and safe the initialization and pointer handling. It enables to build some applications which have the features of modal changing or chance variables elements in real time with iOS native graphic and UI APIs. In the future, we will refer and test the combination the HCI research[12] and generative music.

6. REFERENCES

[1] Wwise, https://www.audiokinetic.com/en/

[2] Adx2, http://www.criware.com/en/products/adx2.html

[3] libPD, https://github.com/libpd/libpd/wiki

[4] AudioKit. http://audiokit.io/

[5] N. J. Bryan, J. Herrera, J. Oh, G. Wang, "MoMu: A Mobile Music Toolkit", *Proceedings of the international conference on new interfaces for musical expression*, Sydney, Australia, 2010, pp. 174-177.

[6] Essl, G. "UrMus - An Environment for Mobile Instrument Design and Performance", *Proceedings of the International Computer Music Conference*, New York, 2010, pp.76-81.

[7] Gibber, http://charlie-roberts.com/gibber/

[8] J. T. Allison, Y. Oh, B. Taylor, "NEXUS: Collaborative Performance for the Masses,Handling Instrument Interface Distribution through the Web", *Proceedings of the international conference on new interfaces for musical expression*, Daejeon, Republic of Korea,,2013, pp.1-6.

[9] A. Ito, K. Watanabe, G. Kuroda and K. Ito, "iSuperColliderKit: A Toolkit for iOS using an internal SuperCollider Server as a Sound Engine" *Proceedings of the 2015 International Computer Music Conference*, Texas, 2015, pp. 234–237.

[10] iSuperColliderKit. https://github.com/wdkk/iSuperColliderKit.

[11] SuperCollider for iOS Sourceforge git repository git://supercollider.git.sourceforge.net/gitroot/supercollider/supercollider isc

[12] K. Kin, B. Hartmann, T. DeRose, M. Agrawala, "Proton: Multitouch Gestures as Regular Expressions", ACM Human Factors in Computing Systems (CHI), 2012, pp. 2885-2894.

The Sky's the Limit: Composition with Massive Replication and Time-shifting

Christopher Coleman
Hong Kong Baptist University
coleman@hkbu.edu.hk

ABSTRACT

Experimentation with tape loops in the 1960's led Steve Reich to develop phase or process music, characterized by immediate and constant repetition of small phrases of recorded speech that are repeatedly replicated and gradually move out of phase with one another. Reich's aesthetic was a practical one, as his control of the phase process was only to decide how many loops to use, when they would enter, and how long the piece as a whole would last. The technology of the time prevented him from being able to control the exact timing of those phase relationships; in his later music for acoustic instruments he varies the phase relationships at much longer time intervals—at regular subdivisions of the prevailing beat. This article describes the development of a compositional method utilizing readily available technology to vastly expand the number of replicated parts and to control the time element of the phase relationship.

1. INTRODUCTION

American composer Steve Reich created a process of composition known as phase music in 1965-66. In two seminal works *It's Gonna Rain*, and *Come Out*, he used a recording of the human voice stored on multiple magnetic tape loops that gradually move out of synchronization with each other. Reich (1974) discovered that by increasing the density of texture and changing relative phase relationships, dramatically new timbres result. The technology at the time limited his ability to multiply the sound source—because of the signal decay of magnetic tape Reich limited himself to eight phase-shifted statements of the original fragment. Furthermore, these two works involve phase relationships whose temporal distances are originally very small—on the level of microseconds—but which are lengthened as the piece progresses. The 'control' of the phase distance is merely the mechanical imperfection of the various tape machines Reich used. In his subsequent music, Reich continued to explore phasing, but without technological assistance, as in *Piano Phase* (1967). He employed a temporal scale measurable by traditional musical notation in works from *Clapping Music* to his Pulitzer Prize winning *Double Sextet*. Phasing in these later pieces is no longer at the microsecond level; it is at the eighth note or sixteenth note, a vastly larger scale. Although Reich's early innovative work has inspired many composers, the use of microsecond phasing as a structural element has had only limited subsequent investigation. Postminimalist composers such as William Duckworth in his *Time Curve Preludes* for piano (1978), and John Luther Adams in his *Dream in White on White* (2009) continue to explore subtle phase relationships on the longer time scales of Reich's later work. Finnish composer Petri Kuljuntausta has frequently explored phase relationships as the basis for entire works—his *Violin Tone Orchestra* (2004) uses short fragments of a sampled repeated violin pitch and repeatedly phases the fragments, but the resultant textures are much simpler than Reich's, due to the simpler source material. More interesting is Kuljuntausta's *When I am Laid in Earth* (2004) that samples two notes from Henry Purcell's *Dido's Lament* and repeats and phases them. Kuljuntausta describes these compositions as "altered music", and certainly *When I am Laid in Earth* bears little resemblance to its source material, as none of the richness of Purcell's harmonies or melodies remain after Kuljuntausta's fragmentation. Kuljuntausta speaks of "frozen sounds" and "microlevel sound phenomena" in reference to his *Four Notes* (2004), a study in the subtlety of timbre. In *Veni Creator Spiritus* (1998), American composer Colby N. Leider highly processes loops of a recording of the Hilliard ensemble singing a Renaissance motet. The phasing, spatialization and other signal processing involved vastly transform the original, but the most significant transformation, and the one rendering the new version fairly unrecognizable, is Leider's fragmentation of the original into bits of only two or three notes. All of these works limit the number of phase relationships to approximately the same as those used by Reich. A somewhat different approach to phase music can be heard in John Oswald's *z24* (2001), in which 24 different performances of the complete and highly recognizable opening fanfare from Richard Strauss's *Also sprach Zarathustra* are superimposed. Rather than the phasing resulting

from looping of a single unique track, it comes from the different lengths of the various interpretations. Lasting not quite two minutes, *z24* contains only near-simultaneous versions, but no subsequent repetitions of the material. Significant aesthetic repercussions result from Zorn's choice. *z24* is in no way a minimalist or post-minimalist piece; in it, phasing is liberated from its minimalist legacy.

Even in this work, the number of simultaneous phase relationships is relatively small. However, commercially available programs such as ProTools, Logic Pro and even Garage Band make available vastly larger potential phase numbers and allow control at microsecond intervals.

2. GENESIS OF A METHOD

I began working with massively replicated and time-shifted music in 2009. Years before, as executor of my parents' estate, I had left the house in which I grew up for the last time while Chopin's Op. 10 No. 1--the Etude in C major--played on the radio. Listening, I was deeply moved, and thought that I must do something with that piece that would capture the panoply of emotions I felt. Some years later I returned to the Chopin with the idea of experimenting with various amounts of time-shifting, inspired by Steve Reich's early phase music. Where Reich relied on the mechanism of the various tape machines running at different speeds, I would use the computer to control time-relationships. One of the most important consequences of the simple program I used was that once the number of replications to be shifted was set, further versions could be added to or deleted, but no analogue 'slippage' of the phase-relationships would occur naturally as it had in Reich's music. The focus of Reich's aesthetic, the constant change of phase relationships between the various replications, would be entirely absent in my work. Instead, I was interested in transforming a much longer span of sound—the entirety of Chopin's etude, not simply a fragment. The etude is a short exercise in arpeggiation, with the right hand sweeping each harmony up and down the keyboard in a relatively slow and fixed harmonic rhythm. The goal was to retain the harmonic implications of the etude while greatly modulating the timbre of the solo piano. Using a pre-existing recording, 16 replications were initially created and time-shifted by a slightly different amount each. By mixing those 16 into a single stereo track, four more iterations of the material were created, using 32, 64, 128, and finally 256 replications, all derived from the mix of the original 16. Rather than programming the time-shifts via a fixed algorithm, I set each one by hand, auditioning the results throughout the process. I experimented with differing time relationships between the parts. Time-shifting varied from extremely short durations—a quarter of a period of a sound wave, to much longer ones—quarters of seconds or longer. (Figure 1)

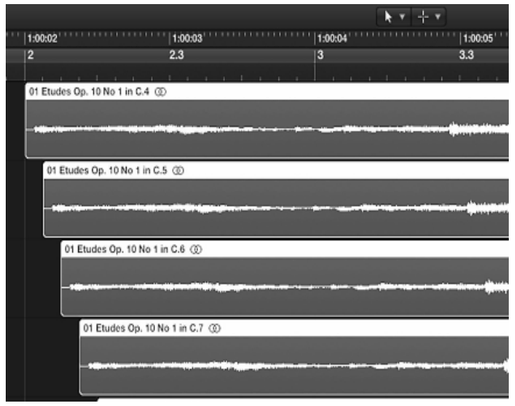


Figure 1. Four of 16 replications of the original Chopin, time-shifted by 1/16th of a second each. (An audio example of all 16 replications can be heard at https://soundcloud.com/christopher-coleman-603014064/icmc-fig-1)

Not only were different time relationships varied, but also different rates at which the time relationship would change between one entrance and the next. These included entrances that were regularly spaced, quasi-exponentially spaced, quasi-randomly spaced, and Fibonacci-series based. (Figure 2)

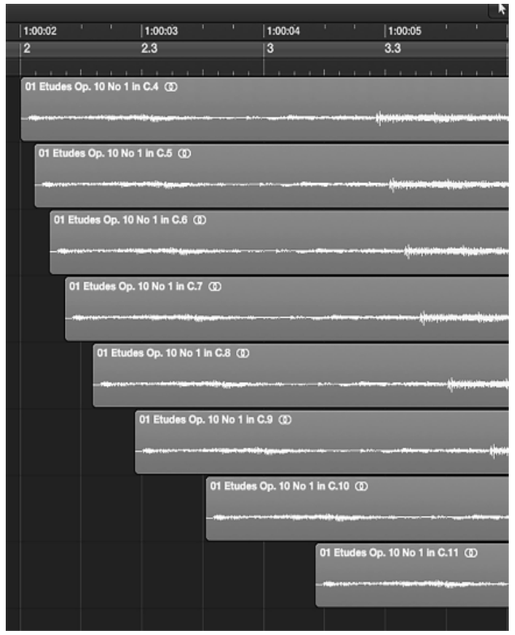


Figure 2. Eight replications, time-shifted in a Fibonacci series in which 1 unit equals 1/8th of a second. (Audio example: https://soundcloud.com/christopher-coleman-603014064/icmc-fig-2)

I was far more attracted to those versions in which the material was time-shifted irregularly. From the highest notes of each arpeggio fascinating rhythmic patterns emerge because of this irregularity. As the number of replications increases, these patterns become ever more complex in a manner somewhat reflective of a Classical variation. The characteristic piano timbre begins to disappear surprisingly quickly and is gone almost completely by the iteration with 64 replications. At this point, only those top notes forming the rhythmic pattern sounded pianistic at all, as they were not masked by the surrounding material. At 128 replications, a very strong sum tone began to emerge in places; beyond 256 this became overwhelming, and the engaging rhythms of the top notes merge into oblivion as well, making the process unusable for my purposes. The final three iterations of the etude, however, transform the piano timbre into a rich organ-like sound as all individual attacks are masked. The choice of Chopin's first etude was felicitous, as the arpeggiation created a natural filter-sweep through each harmony, which I later enhanced digitally.

Structurally the piece is rather simple: after the initial iteration of 16 replications the procedure is repeated with ever-increasing numbers of replications so that the entire etude is heard five times. At this point in the process of composition, the duration (about 10 minutes) seemed satisfactory; the growth of texture achieved the 'other-worldly' transformation inspired by my initial encounter with the Chopin; and the transformation of timbre was fascinating. There were, however, aesthetic problems with the overall structure—five times through the same harmonic material felt wearing to my ear, and the initial iteration consisting of 16 near-simultaneous statements seemed too recognizable and insufficiently transformed. Rather than apply pitch-shifting, which seemed somehow inappropriate, the volume was adjusted so that the piece begins on the very edge of audibility and moves in a series of waves: becoming louder very gradually, disappearing into silence, and emerging again. This obscured the beginning enough and even enhanced the sense of 'other-worldliness' that was so important to the concept of the piece, which I titled *Rainbows, Halos, Glories*. (https://soundcloud.com/christopher-coleman-603014064/rainbows-halos-glories)

Initially I had not conceived of the work for multiple channel playback, but retrospectively considering the resultant music, the match seems almost inevitable. The piece can be diffused a number of ways, with the initial pianississimo music coming from behind the audience (or better, above, if available), further enhancing the distant mystical effect. Depending on the number and location of the speakers, each of the subsequent opening waves can be sent to a different location in the hall. Without resorting to amplitude panning between speakers, a panning effect is created when the various time-shifted tracks are diffused via speakers adjacent to one other. In the final iteration, as the sound literally surrounds the audience, the effect has been described by one listener as "the mother-ship is descending!"

In my second piece to employ this technique, *a multitude, before creation*, from 2011, a pre-existing stereo recording was again used, this time of Giovanni Gabrieli's *Canzon septimi toni,* in a modern rendition for 8 brass instruments. The title refers to the theological paradox that the Bible repeatedly mentions multitudes of angels, yet nowhere in Genesis does God create that multitude—they must therefore have existed before creation, and yet, nothing but God existed before creation. I selected Gabrieli's canzon as it represents to me the absolute pinnacle of abstract religious music. Again I wanted to create a mystical sense, as though this was music of another sphere, and to be referential to the generating piece but not immediately recognizable. The canzon itself is a short work, about three and a half minutes long; time-stretching the original to ten minutes rather than juxtaposing a series of iterations avoided the structural problem of repetition in my earlier piece. This stretching was immediately transformative, not only slowing the tempo dramatically but also granulating the sound. I further wanted to push the number of replications much higher. I created versions in which 16, 256, 4096, 65536, and 1048576 replications were time-shifted (again by hand) by mixing down each of the previous versions to stereo, replicating and phase shifting that stereo mix 16 more times and then time-shifting them.

Each version has its own unique characteristic—the original 16 are time-shifted at rather tight intervals, creating a sound very similar to a large brass ensemble playing in an extremely reverberant hall. The version with 256 replications begins to mask the attacks and blur the harmonies at their changes, but that blurring fades and the harmony eventually resolves as it sustains. The 4096 version takes on a more metallic timbre as the higher harmonics multiply and the bass is attenuated; sum tones occasionally sound strongly here. In the 65536 version, harmonies are completely blurred into a single harmonic field and the metallic timbre and sum tones are more prominent, but wonderfully dramatic registral shifts occur as the tubas enter or drop out. The highly metallic and reverberant 1048576 version, containing over eight million individual brass parts, is scarcely recognizable as being generated by acoustic instruments.

To create the piece, each of these versions were equalized in length (as the higher number of replications naturally takes longer to complete than lower ones), and placed on five concurrent stereo tracks. The volume levels of each track were then individually adjusted, increasing or decreasing freely as the piece progresses. At times only a single track sounds, at other times several tracks together so that, for example, the 16 replications play softly under a somewhat louder 1048576 replications, allowing both the recognizable harmonies and the extreme distortions to be heard simultaneously. Occasionally volume panned somewhat rapidly from one track to another, emphasizing the contours of Gabrieli's textural changes. At the most poignant moment in the piece, where in the original Gabrieli moves from the full ensemble to a single treble instrument, the music moves from the densest, most elec-

tronic-sounding statement to the least dense, most acoustic statement, suddenly clarifying the texture. (Figure 3)

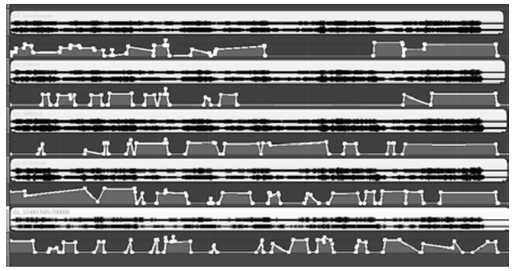


Figure 3. A schematic depiction of the entirety of a *multitude, before creation*, showing amplitude changes between the various tracks. The upper track is a mix of 16 time-shifted replications, the bottom track a mix of over one million. (https://soundcloud.com/christopher-coleman-603014064/a-multitude-before-creation)

My third and fourth ventures into massive replication and time-shifting, *Moro Lasso Loops* from 2012 and *More Moro Lasso Loops* from 2014, turn to vocal music and employ Carlo Gesualdo's *Moro lasso, al mio duolo* as the generating material. The first piece was merely an experiment to determine whether the intensely chromatic harmony of the original 5-part madrigal would prove musically interesting when processed and to get a sense of the effect of replication on the various phonemes. It again used a pre-existing recording. To create the second piece *More Moro Lasso Loops* I rehearsed a quintet of madrigal singers and separately recorded each part, thereby allowing far more freedom in the handling of the material. Each of the voice parts was treated to the massive-replication and time-shifting procedure individually, pushing beyond 2 million replications for each part. I then auditioned each track separately, noticing the most fascinating results, with the intention of combining, for example, 256 basses and tenors with 4096 altos, 1 million second sopranos and 2 million first sopranos. My original thought was to present the madrigal with the same structure and approximate length as the original, about 4 minutes long. As I worked, though, the material seemed so rich and engaging that I reconsidered and treated it far more freely. The beginning and ending remain relatively recognizable, but the middle is completely reorganized, sometimes superimposing the alto of one measure with the soprano of a different measure and a tenor or bass from still other measures. Further, the textural setting of the original is abandoned—if a replicated part is particularly fascinating, it sounds by itself even if it had originally been part of a thicker texture. This mosaic approach was far more complex than any of the previous work and resulted in a seven and a half minute long piece on almost 90 discrete stereo tracks, with video art by Jamsen Law. The original diffusion was a 5.1 mix. The nature of the composition, however, easily allows for effective diffusion over much larger arrays and is in fact far more effective when so performed. (https://soundcloud.com/christopher-coleman-603014064/more-moro-lasso-loops-by-christopher-coleman)

Having experimented with pre-existing compositions, I had learned to anticipate the effects of the procedure on various types of music—the filter sweep effect of rapid arpeggiation, the dramatic consequence of registral change, the heightened sense of anticipation as one harmony blurs into another and slowly resolves. In a series of three *Triptychs* from 2015, each for a different instrumentation, I composed original music designed to exploit these effects. The *Triptychs* all utilize the same structural concept and consist of three movements that can be performed as a suite or separately. The first of these movements is for acoustic instruments alone, the second for fixed media based on massive replication and time-shifting of a recording of the first movement, and the third combines the fixed media with the live instruments from the first movement.

Triptych I is for marimba and almglocken (1 player) and fixed media. The first movement, *Toccata*, begins quietly with the performer striking the marimba keys with his fingers. A single central pitch gradually accumulates other notes and expands into short repetitive patterns before moving into an explosive slapping of the keys with the palms of the hands. The repetitive patterns are then developed in the almglocken before the marimba returns with long arpeggiations that cross the entire instrument. The movement ends with a quiet 4-part marimba chorale with occasional almglocken interpolations. The various sections were composed not only to be successful as a solo piece, but equally importantly, to work when transformed through the technique. The second movement, *Wooden Rain*, takes its title from the sound created when massively replicating the fingertips on the marimba. It loosely follows the structure of the first piece but omits the almglocken passages. Some of the bells had a slight buzz when struck that we could not dampen in the recording studio; when massively replicated that buzz quickly overwhelmed the more sonorous bell timbre and rendered those passages unusable for the effect I was trying to achieve. Other unplanned sounds were more serendipitous. Toward the end of the arpeggiated passage, the percussionist accidentally hit her sticks together; when replicated thousands of times the resultant clatter brings the whole passage to an effective close. When the marimba is played with a hard mallet in the upper register, the sharp attack morphs into a gentle fuzz-like distortion created by the complexity of the high harmonics when replicated massively. Overall, the procedure is deeply transformative of the marimba timbre, often creating a deep pulsing and giving a remarkably human quality to the marimba sound. The movement is mixed onto 50 unique tracks. Naturally, not every track sounds continuously—there is a great deal of spatialized movement rather than a constant envelopment of surrounding sound. (https://soundcloud.com/christopher-coleman-603014064/wooden-rain) The final movement, *Beyond Reality*, combines the massively replicated tracks with the live instruments. The original plan was to have each movement progress through the material in roughly the same order and at the same rate; in practice this proved uninteresting. I abandoned that idea and reorganized the fixed media part by superimposing material in new ways, omitting parts and reshuffling other parts. Ultimately I found this movement less successful than the two previous ones, as the single onstage marimba added very little to the overall sound of the fixed media.

In *Caves of Dunhuang (Triptych III)*, I greatly expanded the number of timbres, composing for erhu, cello, xiao/dizi, clarinet/bass clarinet, yang qin, harpsichord, temple bells and fixed media. As with *More Moro Lasso Loops*, each part was recorded and subjected to the massive replication and time-shifting procedure individually, and time-stretching was used in places. I had noticed that the nature of the technique naturally resulted in a constantly thick and highly reverberant sound. Seeking some contrast, in the second movement of *Caves*, *śūnyatā (emptiness)* (https://soundcloud.com/christopher-coleman-603014064/sunyata-emptiness), massive replications are tempered with some minimally replicated and time-shifted passages. In certain places minimally and maximally replicated instruments sound simultaneously. At one point, the recorded harpsichord and yang qin are replicated in the thousands in quite close time-relationships, creating an active harmonic field, while the recorded cello and bass clarinet, performing contrapuntal lines, are merely tripled in a more relaxed time-relationship. I have further greatly extended the time-frame of overlapping--at one point a harmonic blurring begins that takes an entire minute to resolve. Applying the procedure to the temple bells failed aesthetically—the sharp nature of the attack meant that even the closest phase relationship, when massively multiplied, resulted in an unwanted stutter of multiple attacks rather than a single Ur-bell as desired.

I had felt that the earlier Triptychs had insufficient contrast within and between their movements. To counter this proclivity, and because the material of the Triptychs were originally generated from their first movements, *Caves's* first movement, *madhyamāpratipad (the middle way),* was designed episodically with a great deal of contrast between sections. Different portions of that movement were then used in the latter movements with very little overlap of material. New material was also inserted for the instrumentalists in the third movement, *vijñāna-santāna (rebirth)* (https://soundcloud.com/christopher-coleman-603014064/vijnana-santana-rebirth). Further contrast between movements occurs in the treatment of texture in the fixed media portions. The second movement is far sparser, with only 36 unique tracks, while the third movement has over 90, plus the 6 live instruments. A version incorporating the instrumental parts into fixed media has been created for 124-speaker playback for the 2016 Cube Fest at Virginia Tech.

The technique of massive replication and time-shifting is rich with developmental possibilities yet to be fully explored. A passage of music may sound very different when time-shifted in longer or shorter intervals. Certainly the number of replications sounding nearly simultaneously greatly affects the outcome; exploring the possibilities of the higher number of replications, where timbral transformation is so complete, is rife with potential.

3. SUMMARY

Contemporary technology makes both replication into the millions and time-shifting at microsecond intervals readily available compositional tools. Massive replication and time-shifting offers an effective method of composing high-density music suitable for playback on high-density speaker arrays. The replication/time-shifting procedure has been developed through a series of pieces, initially based on works by other composers but eventually on music composed and designed specifically to take advantage of the effect. Basing an aesthetic on the concept of transforming the original material to an extreme without losing its specific harmonic character creates specific structural problems that can be solved through the creative application of volume control, layering, superimposition and other re-orderings of material.

Acknowledgments

I would like to thank the Research Grants Counsel of the Hong Kong University Grants Committee for support of this paper and the *Triptych* series.

4. REFERENCES

[1] S. Reich, *Writings about music*. Halifax: Press of Nova Scotia College of Art and Design, 1974.

[2] S. Reich, *Piano phase*. London: Universal Ed., 1967.

[3] S. Reich, *Clapping Music*. London: Universal Ed., 1972.

[4] S. Reich, *Double sextet*. London: Universal Ed., 2009.

[5] W. Duckworth, *The time curve preludes.* New York: Peters, 1979.

[6] J. L. Adams, *Dream in White on White*. New Albion Records, NA 061, 2009.

[7] P. Kuljuntausta, *Momentum.* Aureobel, 3AB-0103, 2004.

[8] C. Leider, *Veni Creator Spiritus*. Innova Records, 118, 1998.

[9] J. Oswald, *z24*. Seeland, 515, 2001.

SCATLAVA: Software for Computer-Assisted Transcription Learning through Algorithmic Variation and Analysis

David Su
Inspiro
104 rue d'Aubervilliers, 75019 Paris
david.d.su@gmail.com

ABSTRACT

Transcribing music is an essential part of studying jazz. This paper introduces SCATLAVA, a software framework that analyzes a transcription for difficulty and algorithmically generates variations in an adaptive learning manner in order to aid students in their assimilation and understanding of the musical material and vocabulary, with an emphasis on rhythmic properties to assist jazz drummers and percussionists. The key characteristics examined by the software are onset density, syncopation measure, and limb interdependence (also known as coordination), the last of which introduces the concept of and presents an equation for calculating contextual note interdependence difficulty (CNID). Algorithmic methods for analyzing and modifying each of those properties are described in detail; adjustments are made in accordance with user input at each time step in order to adapt to students' learning needs. Finally, a demonstration of the SCATLAVA software is provided, using Elvin Jones' drum solo from "Black Nile" as the input transcription.

1. INTRODUCTION

Transcription is a fundamental part of the jazz education process, and strengthens multiple facets of musicianship such as ear training, technique, history, and analysis. While the process of learning a jazz transcription is similar to that of learning to play a piece for a classical performance, the end goals vary, as a jazz musician is rarely called upon to recreate a prior performance note-for-note. Instead, the jazz musician aims to assimilate the vocabulary of the performance into his or her own improvisational method [1]. Software tools have proven to be useful for other facets of jazz education [2, 3], but the potential for aiding transcription studies is relatively untapped. This paper presents ***s**oftware for **c**omputer-**a**ssisted **t**ranscription **l**earning through **a**lgorithmic **v**ariation and **a**nalysis* (SCATLAVA), a program that aids with the assimilation of rhythmic material in jazz transcriptions. It uses algorithmic composition and computational analysis to help musicians more efficiently internalize the vocabulary of a transcription as well as learn the music itself with more ease. Given the author's background as a jazz percussionist, the analytical components of the software currently focus on rhythmic properties as applied to drum set performance, although the software can be easily extended to incorporate melodic and harmonic material as well as different target instruments and genre-specific parameters.

2. TECHNICAL OVERVIEW

Figure 1 details the input-process-output (IPO) model used for the SCATLAVA program.[1] The transcribed input data is represented using the platform-agnostic MusicXML format for maximum compatibility across computer systems. Conversion to and from MusicXML is supported by most major notation software such as Sibelius and Finale as well as modern web browsers with libraries such as VexFlow.[2] In addition, the flexible XML tree structure allows other visual elements and metadata, such as titles and annotations, to remain untouched by the parsing process.

Upon initialization of the program, the user can specify b, which represents the number of "bins", or beat windows, of primary or strong beats, that a measure is divided into for analysis purposes. A higher value of b corresponds to increased granularity.

3. PARAMETERS FOR ANALYSIS

Many different methods have been proposed for determining the difficulty of a piece of music, particularly within the realm of music information retrieval [4]. The primary properties of a musical passage currently examined in SCATLAVA are onset density, syncopation measure, and degree of coordination required, denoted by d, s, and c respectively. An onset refers to a non-rest note with nonzero duration, and d can be represented as the number of onsets per strong beat divided by the user-specified granularity. To calculate s, we use a variation on Keith's measure [5], adapted such that strong beats reference the first beat of a bin, and normalized to fit within the framework of variable granularity.

[1] The full source code for SCATLAVA can be found at https://github.com/usdivad/SCATLAVA.
[2] http://www.vexflow.com/
[3] http://www.vexflow.com/

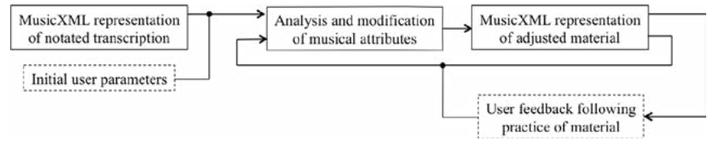


Figure 1: The IPO model for the SCATLAVA program.

Coordination between the limbs, also known as interdependence, refers to when "each limb knows exactly what the others are doing and how they work together, not independently" [6]. Here we present a method for quantifying and calculating c, the degree of difficulty in terms of coordination for a given beat window, resulting in a *contextual note interdependence difficulty* (CNID) value. The equation for calculating the CNID for a beat window is illustrated below:

$$CNID = \begin{cases} 0 + \frac{\alpha_i}{L} \text{ if } n_i = n_{i-1} \text{ and } n_i = n_{i+1} \\ 1 + \frac{\alpha_i}{L} \text{ if } n_i = n_{i-1} \text{ and } n_i \neq n_{i+1} \\ \quad \text{or } n_i \neq n_{i-1} \text{ and } n_i = n_{i+1} \\ 2 + \frac{\alpha_i}{L} \text{ if } n_i \neq n_{i-1} \text{ and } n_i \neq n_{i+1} \end{cases} \quad (1)$$

where n represents the note at subdivision index i of the beat window, α represents the number of simultaneous onsets associated with the note, and L represents the maximum number of simultaneous limbs. By default L is set to 4, representing the use of the left hand, right hand, left foot, and right foot on a typical drum set.

Once d, s, and c have been calculated for a beat window, a weighted average of the three values can be computed to yield a difficulty value D for that period. The precise values of the weights given to each input variable can be adjusted by the user; the program's default weights, denoted by w_p for parameter p, are $w_d = 0.33$, $w_s = 0.33$, and $w_c = 0.34$.

While the value of D for a single beat window can be useful for analyzing that bin itself, the difficulty of an entire measure cannot always be accurately expressed as the mean of its constituent bins' D values. We can see from Table 1 that increasing b yields decreasing values of both s and c but not d; this is due to the fact that onset density is already expressed as a function of b, whereas both the adapted version of Keith's measure and CNID calculation depend on inter-window note onsets. As such,

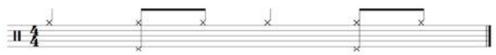


Figure 2: Drum set notation for a basic swing pattern commonly used in jazz music.

the program utilizes $b = 1$ for analysis purposes in order to provide the most comprehensive calculations for s, c, and D. However, as Section 6 details, changing the value of b affects the modifications made to the phrase, and generally values of $b > 1$ yield more musically useful results. Thus, by default $b = 4$ is used when performing adjustments.

4. ADAPTIVE LEARNING

With the computed values of D, we can then begin applying modifications and creating variations on the original transcription in an adaptive learning manner. Adaptive learning refers to a method by which the educator adjusts material presented to the student based on certain properties of how the student is learning [7]. SCATLAVA implements a variant of the method proposed in [8], utilizing user self-assessments to drive its adjustments in order to improve retention of material [9] as well as provide flexibility for the user.

At each time step t, representing the generation, practice, and evaluation of a new score, the user can manually adjust w_p as well as u_p, which denotes the user's confidence value, for each parameter p in [d, s, c]. Each value of u_p is then converted to a gradient, denoted by g_p, which determines the degree to which each difficulty parameter should be adjusted for a given generation. With each successive exercise, the system adapts to the user's learning goals, represented by adjustments in w, and outcomes, represented by values of u.

5. ADJUSTMENT ALGORITHMS

Variations and modifications are made using the following adjustments to yield different values for each parameter: rhythmic expansion or contraction for d, rhythmic transposition [6] for s, and drum set orchestration decisions, such as adding or removing voices and increasing or decreasing repetition of voices, for c. Each of these

b	d	s	c
1	0.25	0.833	0.375
2	0.25	0.667	0.375
4	0.25	0.278	0.167
8	0.25	0.0	0.125

Table 1: Differences in means of d, s, and c, corresponding to a change in b for the measure in Figure 2.

variations can either be created by using the original transcription as input or by a feedback mechanism in which the output of the variation process at step t is then used as input at step $t + 1$. The process of adjustment, which is applied on the scale of each individual bin, is continued until either a target difficulty τ has been reached or a certain number of time steps, adjustable by the user, has gone by without any change in D. The detailed adjustment processes for a single time step are as follows:

5.1 Density of Onsets (*d*)

A single onset from the bin, chosen at random and excluding the first onset in the bin, is removed if the bin contains more than one onset. Figure 3 depicts an example, with the original bin on the left and the two possible outcomes of adjusting d on the right.[3]

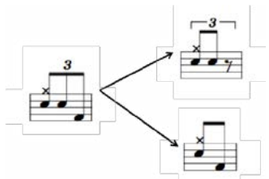


Figure 3: Possible results of adjusting for *d* in a bin.

5.2 Syncopation Value (*s)*

The first onset in the bin is shifted to the beginning of the bin and thus falls on a strong beat according to our syncopation measure. As a result, surrounding syncopations become anticipations, surrounding anticipations become hesitations, and surrounding hesitations are no longer syncopated at all, as seen in Figure 4.

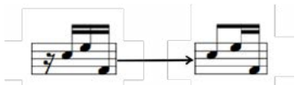


Figure 4: Example of adjusting for *s* in a bin.

5.3 Coordination and Interdependence (*c*)

An onset is chosen such that at least one of the following conditions is true:

I. At least one neighbor of the onset is played on a different surface (i.e. has a different pitch or notehead).

II. The onset is performed with one or more simultaneous onsets.

[3] Note that the bin on the bottom right has been converted to straight eighths in accordance with the jazz notation convention that triplets without the middle note are notated as straight eighths with the understanding that they should be performed swung.

If the onset satisfies condition I, then it is to be played on the surface of the differing neighbor. If both neighbors are different, one of the two is chosen at random. If the onset satisfies condition II, a random simultaneous onset is removed from the bin. If the onset satisfies both conditions, then one of the corresponding actions is chosen at random and applied to the onset. Figure 5 demonstrates the modification possibilities for an example bin when the second note of the bin is selected for adjustment.

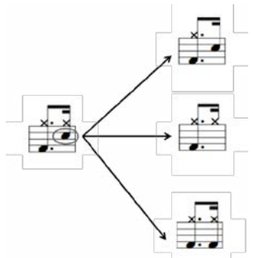


Figure 5: Possible results of adjusting for *c* in a bin, given the selected note (circled above).

For all parameters, if there is no possible adjustment that can be made to a bin, then that bin is returned without any modification. In addition, if the user passes in custom values for each stochastic modifier f_p, then each time an adjustment process is called, the program will use f_p to determine whether the process will actually be executed. The frequency of adjustment for a parameter p is a linear function of g_p.

6. EXAMPLE USING "BLACK NILE"

Elvin Jones' drum solo on the composition "Black Nile" from Wayne Shorter's 1964 album *Night Dreamer* [10] is a popular transcription choice for jazz drummers,[4] especially following educator John Riley's publication of his transcription of the solo [11]. Here the drum solo is used as an example input to SCATLAVA in order to demonstrate the musical output that the program generates. The following examples are selected outputs generated by the program and thus represent a subset of possible outputs given the input parameters used.

In this section SCATLAVA operates on the 4-bar excerpt of the drum solo, transcribed by the author, shown in Figure 6. Upon initial analysis, the phrase yields a dif-

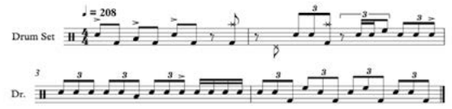


Figure 6: First four bars of Elvin Jones' "Black Nile" solo.

[4] A YouTube search for "elvin jones black nile" yields over 800 results, with videos of other drummers performing transcriptions of Jones' solo comprising 10 of the 14 results on the first page.

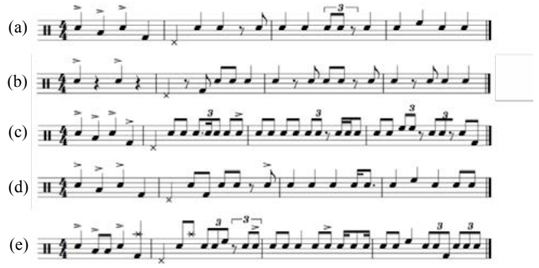


Figure 7: SCATLAVA outputs (using default weights, gradients, and bin divisions unless noted): **(a)** $\tau = 0.2$, **(b)** $\tau = 0.2$, $b = 2$, **(c)** $\tau = 0.2$, $w_d = 0.1$, $w_s = 0.1$, $w_c = 0.8$, $g_d = 0.1$, $g_s = 0.1$, $g_c = 0.8$, **(d)** $\tau = 0.5$, **(e)** $\tau = 0.8$.

ficulty of $D = 0.573$. Default values of $w_d = 0.33$, $w_s = 0.33$, $w_c = 0.34$, and $b = 4$ are used, and the program is initially run with target difficulty $\tau = 0.2$. The resulting output can be seen in Figure 7a. While the contour of Jones' phrasing remains clear, the adjustments render the passage easier to interpret and perform. For example, the first three beats of measure 1 demonstrate the simplification that results in reducing onset density, while the fourth beat exemplifies the reduction in difficulty of both syncopation (the note has been moved from the last eighth note of the measure to the last quarter) and coordination (the simultaneous crash cymbal is omitted).

Figure 7b and 7c demonstrate the versatility of the SCATLAVA software. Perhaps the user would like to see the outline of the phrase on a higher level; by setting $b = 2$ instead of $b = 4$, the resulting output is even less dense than before, though it still maintains the motivic contour of the original phrase. Similarly, it is possible for a student to have little difficulty with onset density and syncopation but to struggle with interdependence. All the user has to do is enter his or her confidence values to reflect that, and the computed gradients will allow the appropriate adjustments to be made; Figure 7c shows the output for parameters $g_d = 0.1$, $g_s = 0.1$, $g_c = 0.8$, with b set to 4 once more. The weights of the parameters have also been adjusted to reflect the gradients. The resulting phrase bears more resemblance to the original passage, with fewer adjustments to density and syncopation, yet the coordination elements are clearly less challenging, and thus present a much lower difficulty to said user.

In general, by increasing the target difficulty we approximate the original transcription more and more closely. Figure 7d shows an example output for $\tau = 0.5$; the most noticeable difference between Figure 7d and Figure 7a is that the former has a higher density of notes. Similarly, Figure 7e shows the output for $\tau = 0.8$, which introduces even more activity across all parameters.

7. CONCLUSION

This paper documented the SCATLAVA software as a model for an adaptive learning environment for the education of jazz transcriptions through algorithmic variation and analysis, with an emphasis on rhythmic material. Together with a variety of user inputs, the software uses onset density, syncopation measure, and limb interdependence using CNID. The primary drawback of the software in its current state is the lack of support for melodic, harmonic, and timbral characteristics. Additional future improvements include more sophisticated machine learning methods to better infer and adapt to users' needs and skills, as well as a streamlined interface that implements methods for learning and playing by ear. Arrangements with jazz educators have been made to begin testing the system with students; this will yield user feedback as well as further results beyond the examples in the paper. With such insights and improvements, it is the author's hope that SCATLAVA will become a powerful yet intuitive software platform for augmenting and extending the tradition of studying transcriptions in jazz.

8. REFERENCES

[1] I. Sandoval Campillo, "In Your Own Sweet Way: A Study of Effective Habits of Practice for Jazz Pianists with Application to All Musicians," Universidad Autònoma de Barcelona, 2013.

[2] C. W. J. Chen, "Mobile learning: the effectiveness of using the application *iReal Pro* to practise jazz improvisation for undergraduate students in Hong Kong," Hong Kong Institute of Education, 2014.

[3] B. Keller, S. Jones, B. Thom, and A. Wolin, "An Interactive Tool for Learning Improvisation Through Composition," *Technical Report HMC-CS-2005-02*, Harvey Mudd College, 2005.

[4] V. Sébastien, H. Ralambondrainy, Olivier Sébastien, and N. Conruyt, "Score analyzer: automatically determining scores difficulty level for instrumental e-learning," *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012*, Porto, 2012, pp. 571-576.

[5] M. Keith, *From Polychords to Polya: Adventures in Music Combinatorics*, Vinculum Press, Princeton, 1991.

[6] J. Riley, *The Art of Bop Drumming*, Alfred Music Publishing, pp. 17-21, 1994.

[7] P. Brusilovsky and C. Peylo, "Adaptive and Intelligent Web-based Educational Systems," *International Journal of Artificial Intelligence in Education*, vol. 13, no. 2-4, pp. 159–172, 2003.

[8] M. Nour, E. Abed, and N. Hegazi, "A Proposed Student Model Algorithm for an Intelligent Tutoring System," *IEEE Proceedings of the 34th SICE Annual Conference*, Hokkaido, 1995, pp. 1327-1333.

[9] P. Sadler and E. Good, "The Impact of Self- and Peer-grading on Student Learning," *Educational Assessment*, vol. 11, no. 1, pp. 1-31, 2006.

[10] W. Shorter, "Black Nile," *Night Dreamer*, Blue Note, 5:02-5:38, 1964.

[11] J. Riley, *The Jazz Drummer's Workshop*, Modern Drummer Publications, pp.43-55, 2005.

Effects of Test Duration in Subjective Listening Tests

Diemo Schwarz, Guillaume Lemaitre
IRCAM–CNRS–UPMC
name.surname@ircam

Mitsuko Aramaki, Richard Kronland-Martinet
LMA–CNRS
surname@lma.cnrs-mrs.fr

ABSTRACT

In perceptual listening tests, subjects have to listen to short sound examples and rate their sound quality. As these tests can be quite long, a serious and practically relevant question is if participants change their rating behaviour over time, because the prolonged concentration while listening and rating leads to fatigue. This paper presents first results of and hypotheses about changes in the rating behaviour of subjects taking a long-lasting subjective listening test evaluating different algorithms for environmental sound texture synthesis. We found that ratings present small but statistically significant upwards tendency towards the end of the test. We put forward the hypotheses that this effect is due to the accustomation of the subjects to the artefacts present in the test stimuli. We also present the analysis of a second test evaluating wind noises in interor car recordings and find similar effects.

1. INTRODUCTION

In perceptual listening tests, subjects have to listen to short sound examples and rate their sound quality. The sound examples would typically be several variants of a speech or sound synthesis algorithm under test, in order to find the best methods or parameters. As these tests can be quite long (usually more than 15 minutes, up to two hours), a serious and practically relevant question is if participants change their rating behaviour over time, possibly because the prolonged concentration while listening and rating leads to fatigue or other long term effects.

This is a real and original research question relevant to countless researcher's daily work, but it is rarely treated specifically in the literature.

We will present analyses of two data sets: A first data set (section 3) with sound quality ratings of five different environmental sound texture synthesis algorithms [1, 2], and a second data set (section 4) from a listening test of unpleasantness of wind noise in car interiors [3].

From an analysis of data set 1, we found that ratings present small but statistically significant upwards tendency in sound quality rating towards the end of the test. We put forward the hypothesis that this effect is due to the accustomation of the subjects to the artefacts present in the test stimuli.

Data set 2 presents a downwards tendency in the pleasantness rating for certain types of stimuli. Here the hypothesis is that listening fatigue could be the main factor.

Of course a good test design would randomise the order of presentation of sounds in order to cancel out these effects for calculation of the mean score for the different stimuli, but they do augment the standard deviation of the results.

2. PREVIOUS AND RELATED WORK

Despite the practical relevance of this question, existing literature on this subject is rather rare. Neither Bech and Zacharov [4], nor Pulkki and Karjalainen [5] treat this question specifically. This observation was corroborated by the reaction of three researchers experienced in designing and carrying out listening test asked by the authors, who all showed surprise at the first hints of an effect. In experimental psychology, Ackerman and Kanfer [7] studied cognitive fatigue in SAT-type tests of 3 to 5 hours, which is much too far from our use case.

We have to look in fields such as usability testing to find relevant research: Schatz et al. [6] study the duration effect of a 90 min test (including a 10 min break) of video transmission quality and web site usability on user ratings. They find little difference of the mean scores of control questions repeated at the beginning and end of the test, although physiological measurements of fatigue (eye blink rate, heart rate mean and variation) and subjective task load index (TLX) questionnaires show clear signs of strain. However, they admit that *"pure audio or speech tests might even cause stronger boredom and fatigue (due to higher monotony) than mixed task profiles"*. Here we can argue further that the mental strain in our experiment 1 is higher, since the *decision rate*, i.e. the number of ratings to decide on is very high—after every stimulus of 7 s, two ratings were required—and more concentrated listening was asked for, whereas in the above studies, rather few judgments from the subjects were required.

We also have to note that above study took place in a lab, and subjects were payed to participate. Our experiment 1 is on-line and unpayed, and the subjects' motivation is thus much lower.

3. EXPERIMENT 1

Data set 1 was collected in a subjective listening test [1, 2], comparing 5 different algorithms for extending an environmental sound texture recording for an arbitrary amount of time, using synthesis based on granular and spectral sound representations, with and without the use of audio descriptors. These algorithms were developed in the course of the *PHYSIS* collaborative research project [1]. Their details are not subject of this article and can be found in [2, 8–12]. See also the state-of-the-art overview on sound texture synthesis [13] for further discussion and a general introduction of sound textures.

The 5 algorithms under test are evaluated in an ongoing listening test accessible online. [2] The experiment setup is briefly described in the following, full details can be found in [1, 2].

3.1 Sound Base

The sounds to be tested stem from 27 original environmental sound texture examples that cover scenes relevant for games and audio–visual applications, such as nature sounds, human crowds, traffic, city background noises, etc. Each original sound of 7 s length is resynthesised by 5 different sound textures algorithms.

3.2 Experimental Procedure

The subjects take the experiment via a web-based form where first the instructions, and then the 27 sounds are presented in random order. For each sound example, the original, and the 6 test stimuli of 7 s length are presented. The stimuli contain in randomised order 5 syntheses, and the original as hidden reference. For each stimulus, the subject is asked to rate the aspects of sound quality and naturalness on a scale of 0–100.

3.3 Experiment 1 Results and Evaluation

Project members and members of the wider research teams were invited by email to take the listening test. There were 17 responders, 16 listening on headphones or earplugs, 1 on studio loudspeakers. None reporting hearing impairments, 5 reported not being familiar with listening tests.

We removed one responder from the statistics (reporting not being familiar with listening tests) who left 80% of the quality and all similarity ratings at the default setting of the web form of 50, and rated the quality of the rest of the stimuli as less than 50.

Figure 1 shows the mean quality and similarity ratings, over all responses and sounds, for the different algorithms. Table 1 shows that the inter-rater reliability, measured by Cronbach's α, is very high (i.e. subjects agree to a high degree in their ratings), with *SDIS* being slightly lower.

	Quality	Similarity
Overall	0.9874	0.9915
Orig	0.9672	0.9789
Descr	0.9431	0.9686
Montage	0.9560	0.9695
AudioTexture	0.9410	0.9628
Random	0.9337	0.9615
SDIS	0.8944	0.8979

Table 1. Inter-rater reliability of experiment 1 (standardized Cronbach's α) for all ratings, and per stimulus type.

[1] https://sites.google.com/site/physisproject
[2] http://ismm.ircam.fr/sound-texture-synthesis-evaluation

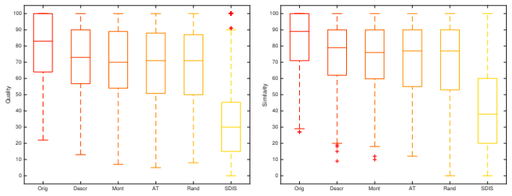


Figure 1. Box plots of the quality and similarity ratings per type of stimulus, showing the median (middle line), quartile range (box), min/max (whiskers), and outliers (crosses).

3.4 Effects of Order on Ratings in Experiment 1

As the perceptual listening test was quite long (the minimal listening time for 27 sounds, each with 6 stimuli and one original, is already $27 \cdot 7 \cdot 7\,\text{s} = 22$ min, the actual test time would be closer to 35 min), the question is if participants change their rating behaviour over time, because the prolonged concentration while listening and rating leads to fatigue.

Figure 2 shows the linear regression fit for all ratings for all synthesised stimuli. The quality ratings show a slight correlation significant at the 1% level with $p = 0.0008$. The slope models a 0.24 and 0.22 increase in quality and similarity rating, respectively, per presentation order.

Figure 3 shows, for each stimulus type, a linear regression fit of the ratings versus the order of presentation of the sound example. We do observe a general trend for the ratings to rise towards the end of the test. For *Descr* and *Random*, the model is significant at the 5% level for quality ratings, for similarity ratings just above 5%, and for *Montage* quality rating at the 10% level. However, only a small fraction of the data is explained by the order, which is good, since we can conclude that the subjects in the test really made an effort to rate the stimuli with concentration and dedication throughout the long perceptual test.

The effect of presentation order is associated to a 0.24 slope that corresponds to a model difference of 6.5 rating points between the first and the last example. For the *Descr* and *Random* quality ratings, we found a 0.27 and 0.28 slope, respectively, that corresponds to a difference of 7.5 points.

Figure 3 also shows the standard deviation of ratings for each stimulus type over order of presentation, and a linear regression fit. These fits show in general a falling trend (the subjects converging towards common values), except for algorithm *SDIS*, which stands out also because it is always rated much lower.

3.5 Hypotheses for Experiment 1

The fact that the rise in ratings is only statistically significant for some of the algorithms, and only for their respective quality ratings, hints at a possible accustomation of the listeners to the artefacts of some of the algorithms.

4. EXPERIMENT 2

Data set 2 is from a psychoacoustic listening test [3] examining the unpleasantness of wind buffeting noises in the interior of 19 car models. The cars were recorded in a wind

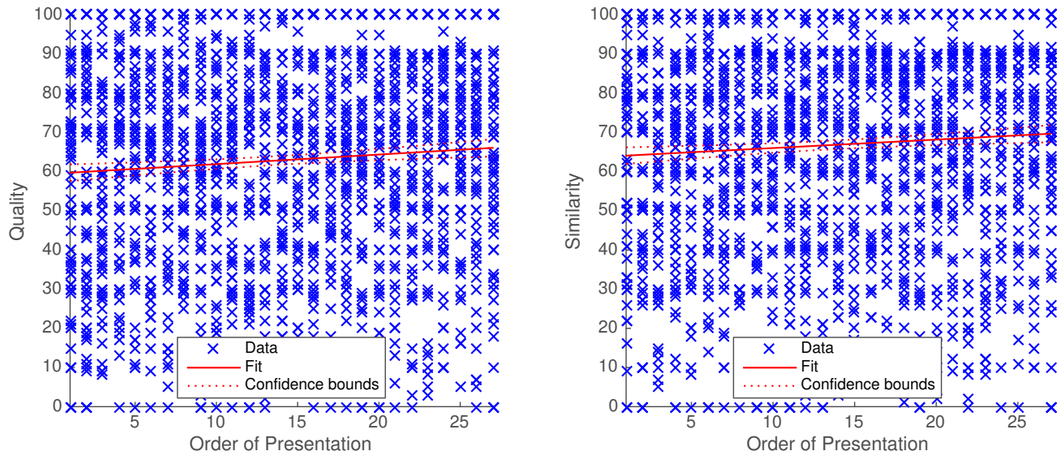


Figure 2. Scatter plots and linear regression fit of all 1215 ratings of experiment 1 for synthesised sounds, explained by order of the sound example. The parameters of the regression models can be found in table 2.

	Quality				Similarity			
	slope	p-value	R^2	adj. R^2	slope	p-value	R^2	adj. R^2
Global	0.24	0.0008	0.51%	0.46%	0.22	0.0030	0.40%	0.36%
Orig	0.11	0.3385	0.21%	-0.02%	0.12	0.2680	0.28%	0.05%
Descr	0.27	0.0366	1.00%	0.77%	0.24	0.0604	0.81%	0.58%
Montage	0.24	0.0673	0.77%	0.54%	0.16	0.2402	0.32%	0.09%
AudioTexture	0.21	0.1057	0.60%	0.37%	0.17	0.2163	0.35%	0.12%
Random	0.28	0.0388	0.98%	0.75%	0.27	0.0516	0.87%	0.64%
SDIS	0.21	0.1537	0.47%	0.24%	0.25	0.1383	0.50%	0.28%

Table 2. Linear regression fit results for experiment 1: slope of the regression line m, p-value of the regression model, and percentage of the variation explained by the model R^2 and adjusted R^2.

tunnel under three different conditions of a buffeting generating device. The test duration was 36 min on average (from 10 to 97 min), and each subject gave 121 ratings in 11 sets of 11 sounds. The experiment design foresaw a lower and an upper anchor reference recording that was present in each set of sounds to rate. In the following we will examine the mean of these ratings only as this eliminates the possibly confounding factors of the 19 different car models and 3 experimental conditions.

Note that the original rating of "unpleasantness" on a range from 0 to 1 has been inverted and rescaled here to a "pleasantness" rating from 0 to 100 to align with the more-is-better valence of experiment 1.

	slope	p-value	R^2	adj. R^2
Global	0.01	0.9127	0.00%	-0.02%
lower anchor	-0.86	0.0001	84.04%	82.27%
upper anchor	-0.12	0.7146	1.56%	-9.38%

Table 3. Linear regression fit results for experiment 2: slope of the regression line m, p-value of the regression model, and percentage of the variation explained by the model R^2 and adjusted R^2.

	α
Global	0.7681
lower anchor	0.9373
upper anchor	0.8900

Table 4. Inter-rater reliability of experiment 2 (standardized Cronbach's α) for all ratings, and per condition.

While the global results in table 3 show that the randomisation evens out the ratings, the regressions for the anchor sounds, visible in figure 4, show no duration effect for the upper anchor, but a highly significant downwards trend of the pleasantness rating for the lower anchor, that makes for a theoretical difference of 9.5 points between the first and last example.

4.1 Hypotheses for Experiment 2

The sound stimuli for this experiment were all real recordings of car interiors, therefore the hypothesis for experiment 1 of accustomation to artefacts of synthesis algorithms can not apply. We hypothesise instead that the downward trend of the pleasantness rating for the lower anchor is due to accumulation of annoyance with the par-

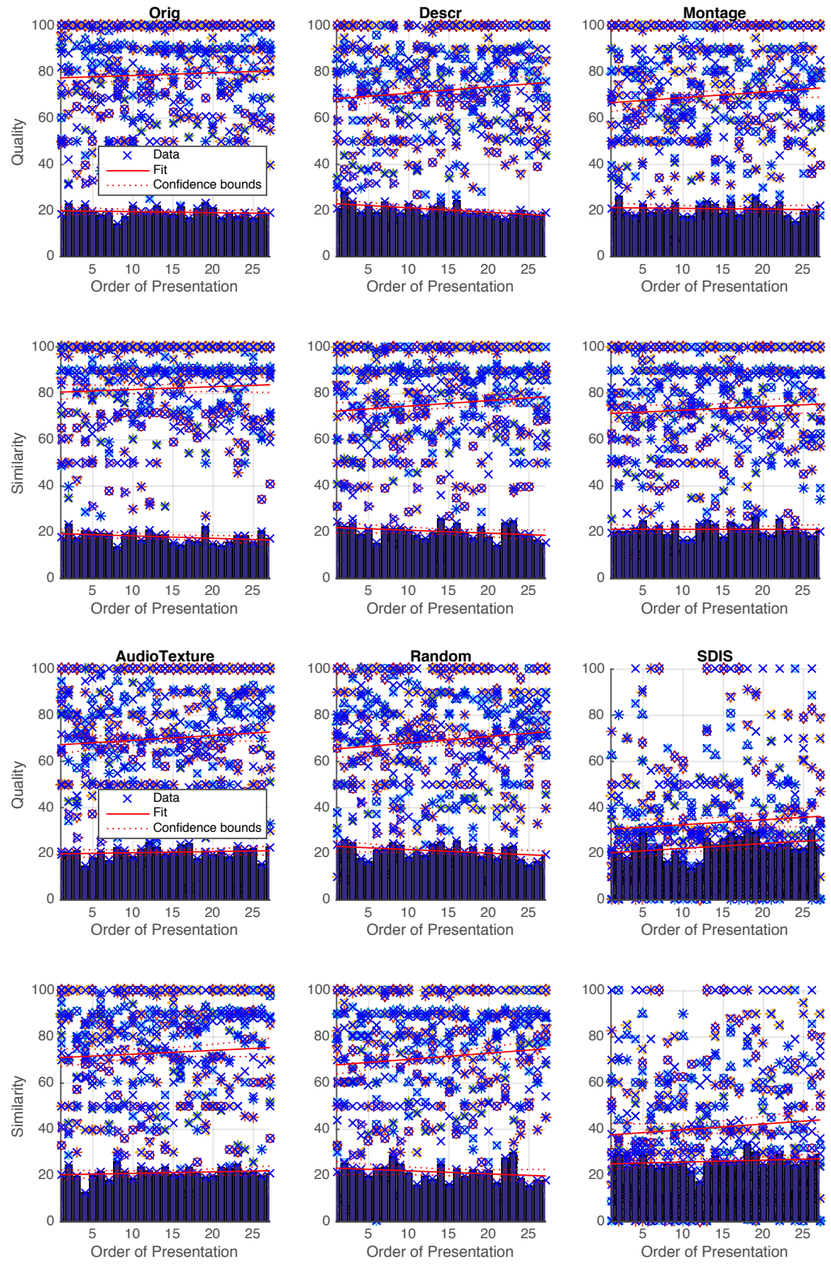


Figure 3. Per-stimulus scatter plots and linear regression fit of ratings of experiment 1 explained by order, overlaid with bar plots of standard deviation and linear regression fit.

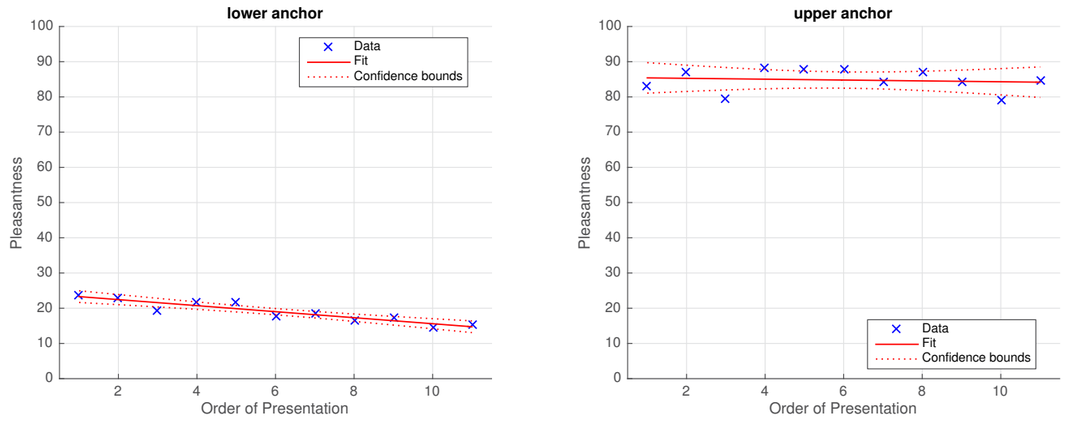


Figure 4. Scatter plots and linear regression fit of the two reference sounds 3 main conditions of experiment 2, rating explained by order of presentation of the anchor sounds. The parameters of the regression model can be found in table 3.

ticularly bad sound of this car, while the upper anchor's much more pleasant sound didn't provoke annoyance in the long term.

5. CONCLUSIONS AND FUTURE WORK

From the analysis of the two data sets we can conclude that there can be effects of changes in the rating behaviour of the subjects in perceptive listening tests over the duration of the tests. These effects vary depending on the type of stimuli and setup of the test. Although randomisation of the order of presentation cancels out these effects for calculation of the mean score for the stimuli, if we were to understand them better, we might reach more contrasted results of the tests, or could devise ways of designing the tests in order to minimise these effects. More research and the analysis of more data sets is necessary to see if the findings presented here generalise to other experiments and setups.

For further work, we could record more precisely the subject behaviour (listening activity and timing), and finally, a closer observation of the physiological and mental state of subjects while taking the test, e.g. via EEG, EMG, or heart rate sensors, could reveal relations between attention signals derived from the sensor data and the hypothesised effects of test duration.

References

[1] The PHYSIS consortium, "Evaluation of example-based sound texture synthesis algorithms of the physis project," *JASA*, 2016, in preparation.

[2] D. Schwarz, A. Roebel, C. Yeh, and A. LaBurthe, "Concatenative Sound Texture Synthesis Methods and Evaluation," in *submitted to DAFx*, 2016.

[3] G. Lemaitre, C. Vartanian, C. Lambourg, and P. Boussard, "A psychoacoustical study of wind buffeting noise," *Applied Acoustics*, vol. 95, 2015.

[4] S. Bech and N. Zacharov, *Perceptual audio evaluation-Theory, method and application*. John Wiley & Sons, 2007.

[5] V. Pulkki and M. Karjalainen, *Communication Acoustics: An Introduction to Speech, Audio and Psychoacoustics*. John Wiley & Sons, 2015.

[6] R. Schatz, S. Egger, and K. Masuch, "The impact of test duration on user fatigue and reliability of subjective quality ratings," *J. Audio Eng. Soc*, vol. 60, no. 1/2, pp. 63–73, 2012.

[7] P. L. Ackerman and R. Kanfer, "Test length and cognitive fatigue: An empirical examination of effects on performance and test-taker reactions." *Journal of Experimental Psychology: Applied*, vol. 15, no. 2, p. 163, 2009.

[8] S. O'Leary and A. Roebel, "A montage approach to sound texture synthesis," in *EUSIPCO*, Lisabon, Portugal, 2014.

[9] ——, "A two level montage approach to sound texture synthesis with treatment of unique events." in *Digital Audio Effects (DAFx)*, Erlangen, Germany, 2014.

[10] W.-H. Liao, A. Roebel, and W.-Y. Su, "On the modeling of sound textures based on the STFT representation," in *Digital Audio Effects (DAFx)*, Maynooth, Ireland, 2013.

[11] W.-H. Liao, "Modelling and transformation of sound textures and environmental sounds," PhD Thesis, Université Pierre et Marie Curie, Jul. 2015.

[12] D. Schwarz and S. O'Leary, "Smooth granular sound texture synthesis by control of timbral similarity," in *Sound and Music Computing (SMC)*, Maynooth, Ireland, Jul. 2015.

[13] D. Schwarz, "State of the art in sound texture synthesis," in *Digital Audio Effects (DAFx)*, Paris, France, Sep. 2011.

The Ear Tone Toolbox for Auditory Distortion Product Synthesis

Alex Chechile
CCRMA, Stanford University
chechile@ccrma.stanford.edu

ABSTRACT

The Ear Tone Toolbox is a collection of open-source unit generators for the production of auditory distortion product synthesis. Auditory distortion products are sounds generated along the basilar membrane in the cochlea in response to specific pure-tone frequency combinations. The frequencies of the distortion products are separate from the provoking stimulus tones and are not present in the acoustic space. Until the release of the Ear Tone Toolbox, music software for the synthesis of auditory distortion products has not been widely available. This first release is a collection of six externals for Max, VST instruments, and patches for the hardware OWL synthesizer, all of which produce various combinations of distortion products and acoustic primary tones. Following an introduction on the phenomenon and an overview on the biomechanics involved, this paper outlines each unit generator, provides implementation examples, and discusses specifics for working with distortion product synthesis.

1. INTRODUCTION

Auditory distortion products (DPs), also known as combination tones or Tartini tones, are intermodulation components generated along the basilar membrane that, under certain conditions, can be perceived as additional tones not present in the acoustic space. Specifically, upon the simultaneous presentation of two frequencies f_1 and f_2 ($f_2 > f_1$) within close ratio (typically 1.22 in clinical settings [1]), DPs appear at combinations of the stimulus frequencies [2], of which the most prominent are f_2-f_1 (the quadratic difference tone, or QDT) and $2f_1$-f_2 (the cubic difference tone, or CDT) [3]. If the stimulus tones are presented through free-field loudspeakers at a moderate to loud amplitude, the resulting DPs can create additional harmonic content and add spatial depth when incorporated in music.

The auditory mechanisms causing DPs are primarily produced in the cochlea. When the cochlea receives sound, the basilar membrane works as a transducer to convey the sound vibrations in the fluids of the cochlea to inner hair cells, which then produce electrical signals that are relayed to the auditory brainstem through the auditory nerve. At the same time, outer hair cells receive electric signals from the brainstem and mechanically vibrate at the frequencies of the sound [4]. This electromotility mechanically increases stimulus-specific vibrations on the basilar membrane, resulting in an increase of hearing sensitivity and frequency selectivity when transmitted to the inner hair cells [5, 6].

However, the outer hair cell movement does not occur exclusively at stimulus frequencies, but is somewhat irregular, thus making its frequency response nonlinear, extending to an audible range [4]. This nonlinear active process increases basilar membrane movement, which aids the loss of energy from damping, while an excess of the generated energy causes additional vibrations that travel backwards from the basilar membrane to the middle ear and the ear canal and creates what is known as otoacoustic emissions [7, 8]. While otoacoustic emissions can be recorded directly in the ear canal with a specially designed earpiece, DPs are specifically the intermodulation components in the inner ear.

It is not surprising that musicians were the first to discover the perception of DPs. Long before the physiological mechanisms behind combination tones were fully understood, musicians Sorge, his colleague Romieu, and Tartini individually found "third tones" produced from two acoustic tones during the middle of the 18th century [3]. In music, evoking DPs can affect the perception of the overall harmony (see Campbell and Greated for an analysis of QDT and CDT in the finale of Sibelius' Symphony No. 1 (1899) [9]). Extending beyond harmonic content, DPs also provide additional spatial depth in music as the acoustic stimulus tones are generated apart from the DPs in the ear, and the DPs are sensed as originating in the listener's head. Composer and artist Maryanne Amacher, known for her use of combination tones in electronic music, discussed the spatial dimension of DPs as a part of a "perceptual geography," and she evoked such environments in immersive compositions and installations [10].

The synthesis of auditory distortion products allows for the precise calculation of perceptual tones which enables the composer and performer access to additional harmonic content, produces spatial depth between sound sources, and creates an intimate interactive listening experience. Until this point, no widely released music software allows the direct synthesis of auditory distortion products. The *Ear Tone Toolbox* (*ETT*) serves to fill this gap, while also operating as an educational tool for hearing DPs and understanding the underlying principles.

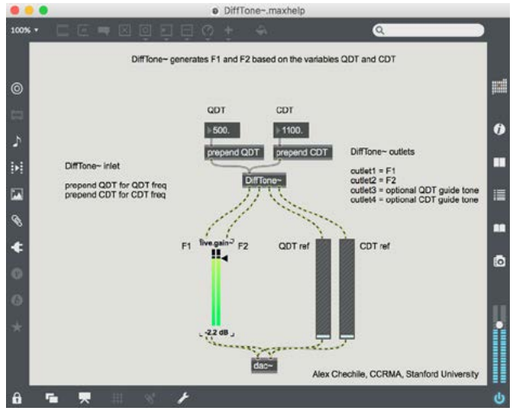


Figure 1. The Max help file for *DiffTone~* provides an overview for the input parameters and output signals of the external object.

2. EAR TONE TOOLBOX

The *Ear Tone Toolbox* is a collection of unit generators for the production of auditory distortion product synthesis. The toolbox generates the necessary acoustic primary tone combinations for evoking perceived DPs, which are not present in the acoustic space. The open-source software was written in the FAUST (Functional AUdio STream) programming language for real-time audio signal processing, and can easily compile to many architectures and formats.[1] In its current state, the *ETT* offers external objects for Max, VST instruments, and patches for the hardware OWL synthesizer.

The toolbox consists of six instruments that allow the user to input various combinations of evoked distortion products and acoustic primary tones. The examples discussed in this paper are in the format of Max external objects. The parameters of each object are sent to the single input, and are specified using the prepend Max object. For example, to control the QDT value, the user passes a numerical value through a prepend object with the argument QDT. The following section provides an overview of each generator in the toolbox.

2.1 Distortion Product Focus with *DiffTone*

The *DiffTone* generator allows the direct synthesis of user defined auditory distortion products. By specifying the desired QDT (f_Q) and CDT (f_C) frequencies, the instrument produces the acoustic primary tones f_1 and f_2 for evoking the distortion products with the equations $f_1 = f_Q + f_C$ and $f_2 = 2f_Q + f_C$.

For example, if a 500 Hz QDT and a 1100 Hz CDT were input, the object would generate two sine tones at f_1=1600 Hz and f_2=2100 Hz. In reverse, we see the two primary tones create the desired combination tones with our original equation for the QDT as 2100-1600 = 500 Hz and the CDT as 2*1600-2100 = 1100 Hz.

[1] `http://faust.grame.fr`

The *DiffTone* generator, along with the other instruments in the toolbox, contains an optional guide tone that can be used for testing, demonstrating, or educational purposes, but otherwise should remain absent from the acoustic signal during normal use. The first and second outlets of the Max object provide the respective f_1 and f_2 sine waves and the third and fourth outlets provide sine wave guide tones at the QDT and CDT frequencies. Figure 1 depicts the Max help file for *DiffTone~*. Like all Max objects in the *ETT*, the input parameters are specified using the prepend object indicating which parameter the user would like to change, which in this case include prepend QDT and prepend CDT.

2.2 Primary Tone Focus with *f1ratio*

Conversely, the *f1ratio* unit generator is primary-tone focused, allowing the user to specify the f_1 frequency and the ratio between the second primary tone. The resulting DPs occur as a byproduct of the given acoustic primary tone f_1 and f_2. Although clinical applications for recording otoacoustic emissions typically use a f_2/f_1 ratio of 1.22, it is possible to achieve more robust DPs at lower interval ratios, with the CDT more sensitive to ratio than the QDT [11]. Hence, *f1ratio* allows for experimentation between stimulus tones as the ratios can dynamically change. The f_2 is calculated with the equation $f_2 = f_1 * r$.

With *f1ratio~* the input parameters are specified with prepend f_1 and prepend ratio, and it produces the f_1 and f_2 sine wave signals from the first two outlets, and the optional guide tones for the QDT and CDT from the third and fourth outlets. The help file for *f1ratio~* is shown in Figure 2.

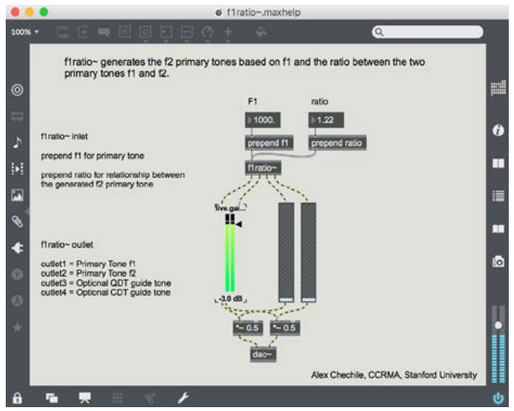


Figure 2. The Max help file for *f1ratio~* illustrates the input and output parameters of the external object.

2.3 Simultaneous DP and Primary Tone Control with *f1half* and *f2half*

For applications where the user requires specific control over both an acoustic component and a distortion product, the following two objects are optimal. The unit generators *f1half* and *f2half* allow the user to specify one of the acoustic primary tones (the f_1 in *f1half* and f_2 in *f2half*) and either the QDT or the CDT. When the QDT is specified, *f1half* calculates the f_2 with the equation $f_2 = f_Q + f_1$, and when the CDT is specified, *f1half* calculates the f_2 with the equation $f_2 = (2f_1) - f_C$. Similarly, *f2fhalf* calculates the f_1 frequency with the equation $f_1 = f_2 - f_Q$ when the QDT is specified, and $f_1 = (f_C + f_2)/2$ when the CDT is specified. It is important to note that both objects require only one specified combination tone while the other must be set to zero (yet the uncalculated second DP will still be produced). The Max help file for *f1half~* is shown in Figure 3. The first two outlets provide the sine wave primary tones f_1 and f_2, and outlets three and four provide optional guide tones for the respective QDT and CDT frequencies.

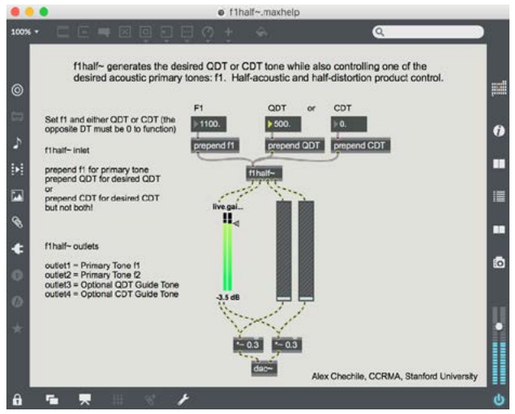


Figure 3. The Max help file for *f1half~*, which is similar to *f2half~*. Note one of the QDT or CDT values in this patch must be set to zero for the object to calculate the second primary tone.

2.4 Distortion Product Spectrum with *DPSpecS* and *DPSpec*

In a series of studies investigating the relationship between combination tones and the missing fundamental, Pressnitzer and Patterson found that the QDT could be perceived at lower primary tone amplitude levels if the stimulus tones were presented in a harmonic spectrum where each primary tone is spaced evenly by a constant value, and the spacing becomes the DP fundamental [12]. Since each subsequent pair of primary tones produces the same QDT, the DP spectrum is perceived at lower amplitude levels due to the vector sum of the various primary tone pairs. The study also found the level of the perceived DP increases with the number of primary tones used. As combination tones and the missing fundamental are produced by different mechanisms, DPs are perceived with primary tones that are in both harmonic and inharmonic relationship to the DP fundamental [13].

The *DPSpecS* unit generator creates a distortion product spectrum following the Pressnitzer and Patterson studies. The user specifies the f_1 acoustic primary tone as well as the DP fundamental f_0, and the synthesizer produces a spectrum of sine waves spaced by the value of the f_0. For example, if the user specifies a f_1 of 1000 Hz and a 100 Hz QDT f_0, the object will output twelve sine waves in stereo (alternating six tones from the first outlet and the other six from the second outlet) spaced by 100 Hz. The third through sixth outlets of the Max object provide optional guide tones for the distortion product fundamental and the next three harmonics. A multichannel version of the instrument, *DPSpec* is also included in the *ETT* and provides individual outlets for each sine wave in the spectrum. The spectrum of primary tones is calculated by $f_n = (n-1)f_0 + f_1$ where f_0 is the distortion product fundamental and $n \geq 2, 3 \ldots 12$.

The QDT between f_2 and f_1 equals the 100 Hz DP fundamental f_0, as does the combination tone between f_3 and f_2, and so forth. The CDT between f_1 and f_2 is also generated (900 Hz with our example), and is emphasized by the subsequent combinations between f_2 and f_4, and again between f_3 and f_6, etc. Distortion products between the harmonics are also produced, although at a lower amplitude. For example, a 200 Hz QDT is produced between f_3 and f_1, and between f_4 and f_2, etc. The Max help file for *DPSpecS~* is found in Figure 4.

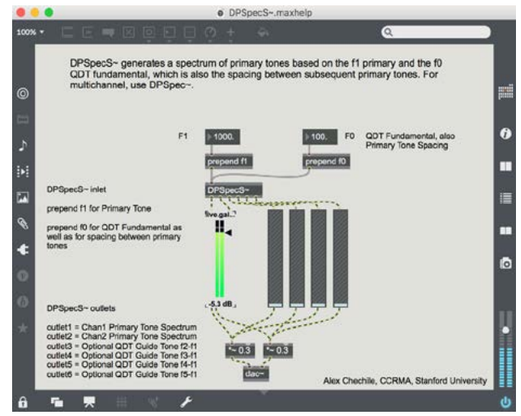


Figure 4. The Max help file for *DPSpecS~* illustrating the input parameters and the output signals.

3. IMPLEMENTATION

Synthesizing distortion products with unit generators enables the user to apply fundamental electronic music techniques[2] for creating larger instrument systems with a high level of creative freedom. For example, Figure 5 depicts a dual auditory distortion product sequencer system in which specific QDT and CDT frequencies can be arranged and manipulated. The two simultaneously running sequencers produce four acoustic primary tones using two *DiffTone~* objects, creating a harmonically complex distortion product spectrum. The two sequences can run in synchronous or asynchronous time, of which the latter produces an evolving spectrum of distortion products. The author uses this technique in his *On the*

[2] For an overview of select DP synthesis techniques, with audio examples, see "Sound Synthesis with Auditory Distortion Products" by Kendall, Haworth, and Cádiz [13].

Sensations of Tone (2010-present) series of compositions [14].

Figure 5. *DiffTone~* implemented in a larger Max patch featuring two sequencers for producing a complex distortion product spectrum.

3.1 On the Sensations of Tone IX: The Descent

On the Sensations of Tone is a series of electronic and electroacoustic pieces that explore the physicality of sound and spatial depth through auditory distortion product synthesis. Presenting multiple acoustic primary tones through multichannel sound systems, the pieces evoke a complex distortion product spectrum while immersing the listener in an interactive sound field where slight head movement causes distortion products to appear, disappear, and change timbre. The structure of each entry alternates between sections that produce DPs and sections that provide contrasting non-DP material. The non-DP sections consist of live or arranged material performed on a modular synthesizer or acoustic instruments.

On the Sensations of Tone IX: The Descent (2015) departs from the aforementioned alternating structure as the majority of the piece is built from field recordings made in the Paris catacombs and the DP material is integrated into the recordings. Using Soundman in-ear binaural microphones[3] and arranged in Ableton Live, the piece relays an auditory narrative of walking through the underground tunnels below Paris. Emerging from the field recordings are two sections built using the multichannel *DPSpec* unit generator from the *Ear Tone Toolbox*. The unique character of the DP synthesis in the piece is the result of integrating the core unit generator into a larger Max patch for further processing. For example, in the second of the two parts containing DPs, the twelve individual sine waves are first amplitude modulated in unison and the resulting individual signals are modulated again in asynchrony. Barely perceivable frequency randomization was applied to both the distortion products as well as the stimulus frequency spectrum. The result of such processing creates an uneven jitter between the primary tones, and evokes fluctuating combinations of DPs.

3.2 Modular Synthesis

In addition to unit generators for computer-based sound synthesis, the *ETT* has been compiled for use on the open-source and open-hardware OWL synthesizer. Built by the London-based collective Rebel Technology, the programmable synthesizer is available as a standalone pedal or as a eurorack module. Both versions contain a STM32F4 microcontroller with a 168 MHz 32bit ARM Cortex M4, 192Kb RAM, 1MB of flash memory and a sampling rate adjustable up to 96kHz.[4] The eurorack version (see Figure 6) allows for control voltage (CV) control over each instrument parameter in the toolbox. The OWL patches are written in C/C++, or the synthesizer can run patches using PureData with Heavy, and FAUST code using faust2owl. An online repository hosts the OWL *Ear Tone Toolbox* patches.[5] Given the form factor of the hardware, the multichannel version of *DPSpec* is unavailable, and the primary tone spectrum is generated with a reduced number of sine tone oscillators. Apart from *DPSpec,* the rest of the instruments in the *Ear Tone Toolbox* run on the OWL similarly to their software counterparts.

Figure 6. The *Ear Tone Toolbox* on the OWL synthesizer eurorack module (bottom row, second from the right).

4. DISCUSSION AND CONCLUSION

Auditory distortion product synthesis offers additional parameters for consideration during music composition and performance and the result provides a unique listening experience for the audience. The direct control over combination tones creates more harmonic and melodic material, which can enhance or intentionally disrupt acoustic material. The distinction between sounds emerging from speakers and other sounds generated within the listener's ears creates an added spatial depth to the work. DPs encourage exploration and interaction with sound as head position and listener location in the acoustic field can produce different combination tones.

After a performance, members of the audience unfamiliar with DP synthesis often report the experience as unlike any other listening experience. The intimacy of allowing a layer of the music to be generated within one's own ears is, however, not for everyone. Audience members suffering from tinnitus or hearing loss occasionally report the experience as disagreeable, or in the latter case, not perceivable.

Aside from the benefits it provides, DP synthesis is not without limitations. The amplitude levels for evoking a strong DP response through free-field speakers often lies within the approximate range of 84-95 dB SPL, which makes reproduction for home recordings or internet distribution difficult. The author prefers to reserve compositions with DP synthesis for concert settings where he can control the amplitude, and make accommodations for the acoustics of the venue. Preparing the audience with a pre-concert introduction on the phenomenon allows the listener to understand the unique aspects of experience. Under controlled conditions, the author finds the majority of the audience's reaction is positive.

The *Ear Tone Toolbox* is the first widely available software package for distortion product synthesis. The six unit generators described in this paper comprise the initial release of the toolbox, with regular updates and additional generators planned. By releasing the software open-source, the author intends to encourage the future development of the field of distortion product synthesis, and to provide educational tools for listening and understanding the fundamentals of combination tones.

Acknowledgments

The author would like to thank Maryanne Amacher, Chris Chafe, Brian Ferneyhough, Takako Fujioka and Pauline Oliveros for their mentorship and support, as well as Cathleen Grado, Romain Michon, and his colleagues at Stanford's Center for Computer Research in Music and Acoustics.

5. REFERENCES

[1] J. W. Hall III, *Handbook of Otoacoustic Emissions*. San Diego, California: Singular Publishing Group, 2000.

[2] H. Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music*, 2nd English ed. New York: Dover Publications, 1954.

[3] R. Plomp, *Experiments on tone perception*. Soesterberg: National Defense Research Organization TNO. Institute for Perception RVO-TNO, 1966.

[4] W. E. Brownell, "Outer hair cell electromotility and otoacoustic emissions," *Ear Hear*, vol. 11, no. 2, pp. 82– 92, Apr. 1990.

[5] T. Gold, "Hearing. II. The Physical Basis of the Action of the Cochlea," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 135, no. 881, pp. 492–498, Dec. 1948.

[6] H. Davis, "An active process in cochlear mechanics," *Hear. Res.*, vol. 9, no. 1, pp. 79–90, Jan. 1983.

[7] D. T. Kemp, "Stimulated acoustic emissions from within the human auditory system," *J. Acoust. Soc. Am.*, vol. 64, no. 5, pp. 1386–1391, Nov. 1978.

[8] D. T. Kemp, *The OAE Story*. Hatfield, UK: Otodynamics, 2003.

[9] M. Campbell and C. A. Greated, *The musicians' guide to acoustics*, 1st American ed. New York: Schirmer Books, 1988.

[10] M. Amacher, "Psychoacoustic Phenomena in Musical Composition: Some Features of a 'Perceptual Geography'," in *Arcana III*, New York, NY: Hips Road, 2008.

[11] J. L. Goldstein, "Auditory Nonlinearity," *The Journal of the Acoustical Society of America*, vol. 41, no. 3, pp. 676–699, Mar. 1967.

[12] D. Pressnitzer and R. D. Patterson, "Distortion Products and the Perceived Pitch of Harmonic Complex Tones," in *Physiological and Psychophysical Bases of Auditory Function*, D. J. Breebart et al., eds., The Netherlands: Shaker Publishing BV, 2001.

[13] G. Kendall, C. Haworth, and R. F. Cadiz, "Sound Synthesis with Auditory Distortion Products," *Computer Music Journal*, vol. 38, no. 4, Winter 2014.

[14] A. Chechile, "Creating Spatial Depth Using Distortion Product Otoacoustic Emissions in Music Composition," presented at the International Conference on Auditory Display, Graz, Austria, 2015, pp. 50–53.

[3] Although recorded in binaural, the final composition with the DPs is intended for concert presentation using free-field speakers.

[4] http://www.rebeltech.org/products/owl-modular/

[5] https://hoxtonowl.com/patch-library/

Sonification of Optically-Ordered Brownian Motion

Chad McKell
Department of Physics
Wake Forest University
chadmckell@alumni.wfu.edu

ABSTRACT

In this paper, a method is outlined for the sonification of experimentally-observed Brownian motion organized into optical structures. Sounds were modeled after the tracked, three-dimensional motion of Brownian microspheres confined in the potential wells of a standing-wave laser trap. Stochastic compositions based on freely-diffusing Brownian particles are limited by the indeterminacy of the data range and by constraints on the data size and dimensions. In this study, these limitations are overcome by using an optical trap to restrict the random motion to an ordered stack of two-dimensional regions of interest. It is argued that the confinement of the particles in the optical lattice provides an artistically appealing geometric landscape for constructing digital audio effects and musical compositions based on experimental Brownian motion. A discussion of future work on data mapping and computational modeling is included. The present study finds relevance in the fields of stochastic music and sound design.

1. INTRODUCTION

In his 1956 work *Pithoprakta* [1], Greek composer Iannis Xenakis modeled a sequence of glissandi after the random walk of Brownian particles in a fluid [2]. Specifically, he assigned values from a Maxwell-Boltzmann distribution of particle speeds to the pitch changes of 46 solo strings. The sequence was unique because it converted an intrinsic aspect of stochastic motion, namely the chance variation in speed between particle collisions [1], to audible sound.

Pioneered by Xenakis, stochastic music represented a slight departure from the indeterminate music written earlier by American composers Charles Ives, Henry Cowell, and John Cage [3]. In *Pithoprakta*, indeterminacy was present in the individual mappings of each instrument, but, as a group, the mappings modeled well-defined laws of probability. In this sense, the composition was both random and deterministic. Although the mappings were also physically-informed, Xenakis appeared to be guided more by statistical descriptions of Brownian motion rather than theoretical diffusion equations or experimental observations of the phenomenon. Moreover, the physical values that defined the motion were modified to accommodate the constraints of a live orchestra. For example, each three-dimensional velocity vector was reduced to a directionless value of speed from which the glissando of an individual instrument could be deduced.

In the present study, an optical trapping setup was implemented in hopes of better harnessing the experimentally-observed nature of Brownian motion for use in data sonification and sound design. An element of determinism was incorporated into the compositional technique by restricting individual stochastic trajectories to user-controlled data-mapping regions. In Section 2, the principles governing the Brownian motion of freely-diffusing and optically-ordered particles are outlined and compared. In Section 3, a historical overview of optical trapping configurations is provided in order to motivate the necessity of a standing-wave optical trap for purposes of surface isolation and data variety. In Section 4, the experimental methods are briefly outlined. In Section 5, the data sonification is described in detail. Finally, the paper concludes with a discussion of future research on data-mapping designs and computational modeling for real-time data sonification.

[1] Although the distribution of speeds was Gaussian, the time intervals used to define the speeds, or glissandi of each instrument, appear to have been imposed arbitrarily [2, Fig. I–7].

2. BROWNIAN MOTION

2.1 Freely-Diffusing Particles

The equation of motion for the finite trajectory of a freely-diffusing Brownian particle of mass M in a uniform viscous fluid is [4]

$$M\ddot{\mathbf{r}} \;=\; -\gamma\dot{\mathbf{r}} + \sqrt{2\gamma k_{\mathrm{B}}T}\,\Gamma(t), \tag{1}$$

where γ is the fluid drag coefficient, k_{B} is Boltzmann's constant, T is the temperature of the fluid, Γ(t) is the zero-average Gaussian white noise, and $\mathbf{r} = x(t)\hat{x} + y(t)\hat{y} + z(t)\hat{z}$ is the particle's position as a function of time t. By "freely-diffusing", it is understood that the particle's trajectory is only determined by the molecular interactions with the background medium, assuming the walls of the fluid chamber are sufficiently far away from the particle and evaporation of the fluid is negligible. Since the inertial term $M\ddot{\mathbf{r}}$ is small compared to the drag term $\gamma\dot{\mathbf{r}}$ in a viscous fluid, the inertial term can be dropped from Eq. (1). This simplification gives the following solution [2] for the velocity $\dot{\mathbf{r}}$ of the particle as a function of time t:

$$\dot{\mathbf{r}} \;=\; \sqrt{2D}\,\Gamma(t). \tag{2}$$

[2] The system described by Eq. (2) is said to exhibit "overdamped" behavior since the viscous damping overpowers the inertial acceleration.

Here, D is the theoretical diffusion coefficient defined by Einstein's formula:

$$D \;=\; \frac{k_{\mathrm{B}}T}{\gamma}. \tag{3}$$

Absent any external force, the particle will amble indefinitely through the fluid in an indeterminate manner. A sonification scheme based on the experimental position data of a freely-diffusing Brownian particle, as characterized above, is limited in at least three ways: (1) the range of data values is generally unpredictable; (2) the collected data will be sparse given the experimental state of the art for imaging unbounded Brownian microparticles; and (3) the data will be limited to two dimensions, barring the use of a sophisticated method for measuring vertical displacement from the imaging plane. In other words, there is an undesirable indeterminacy on the physical range of values obtained for mapping the audible parameters as well as constraints on the size and dimensions of the data. In an ideal scenario, however, the composer would have some control over the range of values along with ample and varied data to choose from.

To remedy these shortcomings, the particle's motion was confined to manageable regions of study by adding an optical-trapping potential $V(\mathbf{r})$ to the system.

2.2 Optically-Ordered Particles

Inserting the diffusion coefficient D and the trapping potential $V(\mathbf{r})$ into Eq. (1) gives the following time-varying solution for the velocity $\dot{\mathbf{r}}$ of an optically-trapped (i.e. "optically-ordered") Brownian particle in a viscous fluid:

$$\dot{\mathbf{r}} \;=\; -\frac{\nabla V(\mathbf{r})}{\gamma} + \sqrt{2D}\,\Gamma(t). \tag{4}$$

As in Eq. (2), the inertial term $M\ddot{\mathbf{r}}$ was omitted to reflect the overdamped nature of the motion. The confining potential $V(\mathbf{r})$ is defined by [5]

$$V(\mathbf{r}) \;=\; -n\alpha\frac{|\mathbf{E}(r,z)|^2}{2}, \tag{5}$$

where n is the refractive index of the background viscous fluid, α is the polarizability of the particle, $r = \sqrt{x^2+y^2}$ is the particle's lateral displacement from the z-axis, and $|\mathbf{E}(r,z)|^2$ is the magnitude squared of the total electric field of the optical beam. The total field irradiance $I(r,z)$ of the beam is proportional to the total electric field by the relation [5]

$$I(r,z) \;=\; \frac{\varepsilon_0 c_0}{2}|\mathbf{E}(r,z)|^2, \tag{6}$$

where ε_0 is the electric permittivity of free space and c_0 is the speed of light in vacuum. Along a two-dimensional trapping plane of the optical field, the solution for the velocity $\dot{\mathbf{r}}$ of the particle becomes

$$\dot{\mathbf{r}} \;=\; \frac{n\alpha\nabla_{\perp}I(r,h)}{\varepsilon_0 c_0\gamma} + \sqrt{2D}\,\Gamma(t), \tag{7}$$

where $\nabla_{\perp}I(r,h)$ is the transverse irradiance gradient of the laser beam along the $z = h$ trapping plane. The following finite-difference algorithm can be implemented to solve this stochastic differential equation numerically for the positions $\mathbf{r}_i = [x_i, y_i, h]$ as a function of the times $t_i = i\Delta t$ [6]:

$$\mathbf{r}_i \;=\; \mathbf{r}_{i-1} + \frac{n\alpha\nabla_{\perp}I(r,h)\cdot\mathbf{r}_{i-1}}{\varepsilon_0 c_0\gamma}\Delta t + \sqrt{2D\Delta t}\,\mathbf{w}_i. \tag{8}$$

Here, i is the iteration of the finite-difference simulation, Δt is the time step, and $\mathbf{w}_i$ is a vector of Gaussian random numbers with unit variance and zero mean. In the next section, a brief overview of the historical development of optical traps is provided in order to elucidate the advantages of using a standing-wave optical trap to analyze Brownian motion compared with other trapping models.

3. AN OPTICAL TRAPPING ODYSSEY

3.1 Particle Acceleration and Confinement

The acceleration of matter by radiated light pressure was first explained by Johannes Kepler in 1619 [7]. Due to the immense irradiance of light emitted by the sun, Kepler observed that the gas and minerals of a comet could be pushed by the light. In 1873, James Maxwell discovered that the radiated light pressure P was equal to the time-averaged field irradiance I of the light divided by the speed of light c [8]. In theory, light radiation pressure could be used to accelerate particulate matter on Earth, assuming the irradiance of the light was substantially large compared to the magnitudes of the perturbed masses.

With the invention of high-irradiance lasers in 1960 [9], light radiation pressure could be feasibly applied to the acceleration and confinement of microscopic-sized particles. The first laser trap was developed in 1970 by Arthur Ashkin at Bell Laboratories [10]. It consisted of two counter-propagating, coaxial Gaussian beams focused at points upstream from their plane of intersection, as shown in Fig. 1. A microsphere located inside the optical field of the two beams was pulled toward the propagation axis (i.e. z-axis) by a transverse gradient force $\mathbf{F}_{\mathrm{grad}}$ and accelerated downstream by an axial scattering force $\mathbf{F}_{\mathrm{scat}}$. Together these forces tightly confined the particle at the point where the intersection plane of the beams met the propagation axis. A breakthrough in laser technology, Ashkin's trap eventually inspired the development of a wide array of trapping configurations, including optical tweezers [3].

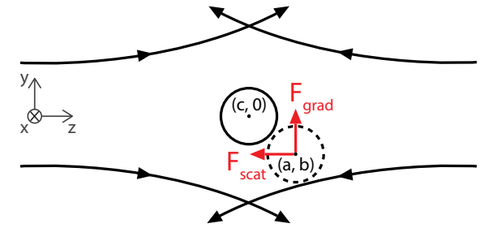


Figure 1. The first optical trap. Two opposing laser beams intersect along the $z = c$ plane. A spherical particle located at the point (a, b) in the zy-plane is pulled to the force equilibrium position $(c,\ 0)$ by optical forces $\mathbf{F}_{\mathrm{grad}}$ and $\mathbf{F}_{\mathrm{scat}}$. Note: The positive x-axis is into the page.

[3] One of the most common optical trapping designs employed by scientists today, optical tweezers are optimal for high-precision, three-dimensional manipulation of microscopic particles.

Although these early trapping designs provided a useful means for stable, surface-isolated trapping, they were not ideal for tracking Brownian motion because they either eliminated the particle's microscopic motion or complicated the imaging process. Standing-wave traps, on the other hand, allowed for enlarged, two-dimensional trapping regions that were convenient for analyzing Brownian diffusion. Additionally, the particles could move vertically from one trapping level to another, permitting a more diverse collection of data.

3.2 The Brownian Trap

The Brownian trap is a standing-wave optical trap containing a vertical lattice of individual trapping regions that are ideal for tracking transverse particle diffusion. The first standing-wave optical trap was developed in 1999 by Zemánek et al. [11]. In a typical standing-wave trapping setup, a laser beam reflects off a mirrored surface positioned perpendicular to the propagation axis and superimposes on the incident beam. The superposition of the beams produces an optical standing wave capable of simultaneous particle confinement in separate, surface-isolated [4] regions.

When fluid-immersed microspheres are introduced in the vicinity of the laser trap, optical forces pull the spheres toward the antinodes of the standing wave, enclosing them in two-dimensional optical pockets [5]. Assuming the counter-propagating beams of the standing-wave trap are well-aligned, the spheres primarily [6] experience an axial gradient force $\mathbf{F}_{\mathrm{X}}$ and a transverse gradient force $\mathbf{F}_{\mathrm{grad}}$. By analogy to gravity, the optical barriers induced by $\mathbf{F}_{\mathrm{X}}$ and $\mathbf{F}_{\mathrm{grad}}$ confine a single microsphere along a particular antinode like a marble in a bowl, as depicted in Fig. 2. Due to molecular interactions with the fluid, the particle may jump in and out of the trap. However, the barriers tend to contain the motion within the optical field.

With test particles captured in the confinement regions of the Brownian trap, one can record the positions of the particles over time using experimental imaging and tracking tools. The tracked points can then be mapped to audible parameters to create a data sonification of experimental Brownian motion.

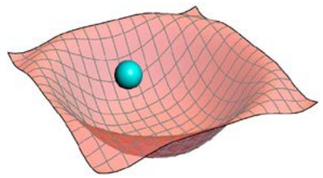


Figure 2. Force field analogy. The optical force field encountered by a microsphere at an antinode of the Brownian trap is similar to the gravitational field experienced by a marble rolling in a bowl.

[4] Surface isolation simplifies the motion by eliminating surface drag.

[5] Refer to Fig. 4 for an illustration of a standing-wave optical trap. Note that the trapping regions (dotted lines) lie along the antinodal planes.

[6] The reflective surface may transmit some laser light for imaging purposes. In such cases, a net axial scattering force $\mathbf{F}_{\mathrm{scat}}$ oriented downstream is also present in the trap, shifting the trapping planes slightly downstream from the antinodal planes.

4. EXPERIMENTAL SETUP

To obtain tracking data of experimental Brownian motion, fluorescent microspheres were inserted into the optical field of a Brownian trap [7]. The particles were imaged with a CCD camera at a rate of 15 frames per second. Video files of individually trapped particles were analyzed using video tracking software in order to determine the horizontal (x, y) positions of each particle over time. The data sets ranged from 98 to 1690 points. The horizontal magnitudes r_i of the displacement vectors $\mathbf{r}_i$ at each time t_i were subsequently calculated. The vertical displacements z_i were determined based on the sizes of the diffraction patterns produced by the diffusing fluorescent spheres relative to a measured standard [8]. In the following section, the data-mapping scheme used to sonify the horizontal and vertical displacements is outlined. Web addresses containing audio samples of the sonified data are also provided.

5. DATA SONIFICATION

5.1 Audio Samples

To hear samples of the data sonification, email the author or visit brownian.bandcamp.com. Audio-visual samples are also available online at youtube.com/chadmckell and vimeo.com/brownian.

5.2 Horizontal Dynamics

5.2.1 Equal-Area Mapping

The radial displacements r_i of individual Brownian particles were mapped to specific notes on a selected musical scale for every time t_i (see Fig. 3). To sonify the horizontal data, two data-mapping approaches were implemented—equal-area and biased mapping. In equal-area mapping, the total area [9] of the trapping region was divided into sub-regions of equal area $A_{\mathrm{E}} = \pi/8\ \mu\mathrm{m}^2$, as plotted in Fig. 3 (middle). Each sub-region corresponded to a unique MIDI note number m_i on a particular scale. In the chromatic scale on C, for example, an r value in the range $(0 \le r < \sqrt{8}/8)\ \mu$m mapped to C4 ($m = 60$); an r value in the range $(\sqrt{8}/8 \le r < 1/2)\ \mu$m mapped to C#4 ($m = 61$); and so forth.

Mapping algorithms were programmed in Java to determine the MIDI note numbers m_i for every displacement r_i measured at time t_i. The computed note arrays $\{m_1, m_2, m_3 ...\}$ were then inserted into Pure Data (Pd) and sampled at a rate [10] of 15 Hz (900 beats per minute). Reverberated sine waves were generated using objects `osc~` and `freeverb~` in Pd. Files containing the MIDI note arrays and Pd patches are available from the author on request.

[7] A detailed description and analysis of the laboratory setup is forthcoming [5]. Information is included about the physical parameters of the laser and other optical components, the measured distances between the components, the sizes and material composition of the spheres, and the experimental tools used to collect and analyze videos of trapped particles.

[8] See [5, pp. 57–61].

[9] The total area of the trapping region varied depending on the maximum horizontal displacement from the origin.

[10] A sampling rate of 15 Hz was chosen in order to match the frame rate of the imaging camera.

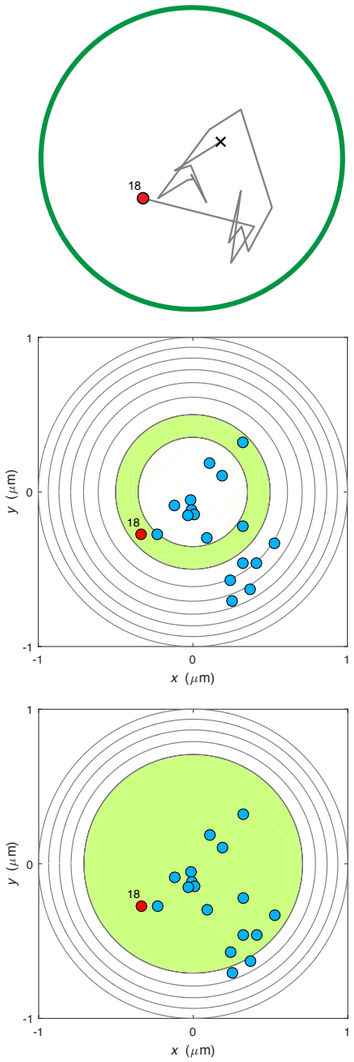


Figure 3. Data-mapping scheme—horizontal dynamics. Top: Each radial displacement r_i of a trapped microsphere mapped to an audible pitch. As the sphere moved from its starting point ("×") to its ending point ("18") in 18 steps, the pitch was updated 18 times at a sampling rate of 15 Hz. Middle (*equal-area mapping*): The total area of the trapping region was divided into sub-regions of equal area. In the example depicted here, the radial points in each sub-region mapped to a unique MIDI note number m_i in the chromatic scale on C. The ending point ("18") mapped to C#4 since it was located in the second sub-region (shaded area) from the origin. Bottom (*biased mapping*): the area of the centermost sub-region was increased to encircle the majority of the radial points. The new mapping reassigned the ending point ("18") to the centermost sub-region (shaded area) so that the point charted to C4.

5.2.2 Biased Mapping

Biased data mapping allows the composer to increase the stability of the centermost (i.e. lowest-frequency) note while retaining the stochastic nature of higher-pitched note combinations. In this mapping approach, the area of the centermost sub-region is enlarged in order to increase the probability that a given r value will lie within the centermost sub-region. In Fig. 3 (bottom), the area of the centermost sub-region was increased to $A_{\mathrm{B}} = \pi/2\ \mu\mathrm{m}^2$ so that, in the chromatic scale on C, an r value in the range $(0 \le r < \sqrt{2}/2)\ \mu$m mapped to C4. Each remaining sub-region retained an area of $A_{\mathrm{E}} = \pi/8\ \mu\mathrm{m}^2$ so that an r value in the range $(\sqrt{2}/2 \le r < 2\sqrt{10}/8)\ \mu$m mapped to C#4; an r value in the range $(2\sqrt{10}/8 \le r < \sqrt{3}/2)\ \mu$m mapped to D4; and so forth.

5.2.3 Multiple Particles

Single particle sonifications were summed in Pd in order to hear the individual stochastic trajectories en masse. Chordal harmonies were created by charting the centermost sub-region of each trajectory to a different note in a chosen scale. While dissonant, equal-area mapping more accurately portrayed the movements of each particle near the origin of the tracking grid. Biased mapping, on the other hand, allowed for more musical consonance.

5.3 Vertical Dynamics

The three-dimensional [11] nature of the tracking data emerged when transitions between trapping planes were measured. As the fluorescent microspheres moved from one trapping pocket to another, fluorescent light from the spheres diffracted through the imaging apparatus. The sizes of these diffraction patterns were compared to a measured standard in order to determine the discrete, vertical displacements z_i of the particles. One possible mapped trajectory of a particle moving vertically in the trap is illustrated in Fig. 4.

To sonify the vertical jumps z_i, the mapping region (i.e. the range of mapped notes) was shifted by one octave for every unit transition along the lattice. For example, if a microsphere dropped down two trapping levels, the mapping region transposed down two octaves; if the particle jumped up one trapping level, the region moved up one octave.

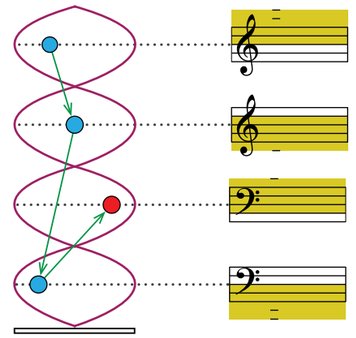


Figure 4. Data-mapping scheme—vertical dynamics. A transition of one antinode along the standing wave transposed the mapping region (shaded area on the staff) by one octave. In the scenario shown here, the trajectory caused a shift of one octave down, then two octaves down, then one octave up. Note: each mapping region in this example spanned one octave in the chromatic scale on C. In practice, however, a typical mapping region covered several octaves.

[11] The three dimensions are represented by the cylindrical coordinates r, θ, and z. Mapping the polar coordinate θ is reserved for future work.

6. FUTURE DIRECTIONS

6.1 Data Mapping

In order to extract more artistic value from the experimental tracking data discussed in this paper, future work on data mapping is proposed. The data-mapping scheme outlined in Section 5 is one among many possible methods. In addition to pitch, the calculated displacements [12] may be mapped to other audible variables, such as timbre and amplitude panning. In two-dimensional vector base amplitude panning (VBAP) [12], the gain factors g_i of individual loudspeakers in a circular array could fluctuate in accordance with a particle's position inside a trapping region, as depicted in Fig. 5. Assigning different values to the sampling rate [13] and the mapping areas A_{E} and A_{B} may also be explored. Additionally, other optical trapping setups, aside from standing-wave traps, could be devised. Maximizing the complexity of these trapping configurations would be particularly desirable since a more intricate setup would lend more options for mapping the data.

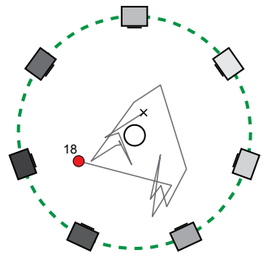


Figure 5. Two-dimensional VBAP. A listener (white circle) perceives higher gain factors g_i (darker shading) from loudspeakers located closer to a particle's mapped position ("18"). Note: the trapping region (dashed border) was scaled to match the size of the circular loudspeaker array.

6.2 Computational Modeling

Given values for each of the physical variables in Eq. (8), a computer program can be written to generate a continuous stream of Brownian position data for real-time sonification and manipulation. In a model based on the equation of motion of an optically-ordered Brownian particle, the composer would adjust the parameters of the "Brownian" audio effect by altering the physical parameters of the particle, the Brownian trap, or the background fluid environment. Increasing the transverse irradiance gradient $\nabla_{\perp} I(r, h)$ of the laser, for example, would make the particle more likely to reside in the centermost sub-region and less likely to escape the trap. Such a change would increase the stability of the lowest-frequency note and reduce the likelihood of higher-pitched, stochastic sequences. Increasing the fluid viscosity γ, moreover, would slow the particle's average velocity, effectively increasing the duration over which notes were played. Data-mapping algorithms could also be incorporated into the model to manipulate the sampling rate and mapping areas in real time.

[12] Apart from displacement, other physical observables, such as average velocity, may be computed from the tracking data and then sonified.

[13] Although the sonification would no longer accurately reflect the physical scenario observed in the laboratory, changing the sampling rate to mismatch the imaging rate may be of artistic interest.

7. CONCLUSION

Data sonifications based on optically-ordered Brownian motion benefit from the fact that the data range can be controlled and the data size and dimensions can be maximized. In this study, a standing-wave optical trap was used to restrict the random motion of diffusing Brownian microspheres to an ordered stack of two-dimensional trapping planes. It was shown that the arrangement of the particles in the lattice provided an attractive framework for producing diverse and manageable sonification data. Lastly, a discussion of potential avenues for research in data mapping and computational modeling was included in order to propel the ideas outlined in the paper.

8. ACKNOWLEDGMENTS

The author would like to thank Keith Bonin for guiding the experimental work that inspired this study; Research Corporation and Wake Forest University for funding the experiments; and David Busath, Justin Peatross, Steve Ricks, Christian Asplund, Rodrigo Cádiz, Kurt Werner, Nick Sibicky, Adam Brooks, and Katherine McKell for discussing ideas and offering feedback about the project.

9. REFERENCES

[1] I. Xenakis, *Pithoprakta*. Boosey & Hawkes, 1967.

[2] ——, *Formalized Music: Thought and Mathematics in Composition*. Pendragon Press, 1992.

[3] P. Griffiths, "Aleatory," *The New Grove Dictionary of Music and Musicians*, vol. 1, 2001.

[4] D. Gillespie and E. Seitaridou, *Simple Brownian Diffusion: An Introduction to the Standard Theoretical Models*. Oxford University Press, 2012.

[5] C. McKell, *Confinement and Tracking of Brownian Particles in a Bessel Beam Standing Wave*. Master's Thesis, Wake Forest University, 2015.

[6] G. Volpe and G. Volpe, "Simulation of a Brownian particle in an optical trap," *American Journal of Physics*, vol. 81, no. 3, pp. 224–230, 2013.

[7] J. Kepler, *De cometis libelli tres*, 1619.

[8] J. C. Maxwell, *A treatise on electricity and magnetism*, 1st ed. Clarendon Press, 1873.

[9] T. H. Maiman, "Stimulated optical radiation in ruby," *Nature*, 1960.

[10] A. Ashkin, "Acceleration and trapping of particles by radiation pressure," *Physical Review Letters*, vol. 24, no. 4, pp. 156–159, 1970.

[11] P. Zemánek, A. Jonáš, L. Šrámek, and M. Liška, "Optical trapping of nanoparticles and microparticles by a Gaussian standing wave," *Optics Letters*, vol. 24, no. 21, pp. 1448–1450, 1999.

[12] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, 1997.

Cybernetic Principles and Sonic Ecosystems

Daren Pickles
Coventry University
d.pickles@coventry.ac.uk

Adam Collis
Coventry University
a.collis@coventry.ac.uk

ABSTRACT

The theoretical basis for the installation Oscilloscope *is discussed in this paper along with a description of the applications of these ideas in the practical implementation of the work. It is argued that, despite the different idioms these practitioners work in, there are conceptual commonalities in the generative music of Brian Eno and the musical ecosystems of Agostino Di Scipio. Both these artists' work is influenced by principles of cybernetics, in particular the notion of emergence where the composer's role is not on designing outcomes but on designing systems whose component interactions produce desirable outcomes. A synthesis of these ideas are also applied in the design of* Oscilloscope*, demonstrating how a system that is relatively simple technologically and with fairly trivial sonic and visual material can be tuned to produce interactions that generate complex results that provide a rich, engaging experience for the viewer. In addition, this discussion critiques the notion of interactivity in electronic music.*

1. INTRODUCTION AND SYSTEM DESIGN

An installation, by necessity, establishes a relation to its setting which, either through reinforcement or contrast, reveals conditions or characteristics of the environment and indeed of the artwork itself. It can therefore be said that installations, to some degree, interact with their environmental setting. Where installations contain dynamic, non-corporeal phenomena, such as sound and video, deeper forms of interactivity are afforded.

Oscilloscope is an installation featuring sound and computer animations generated in real time in response to image data of the installation's environment captured from a camera. This work was developed for the launch night of the city of Coventry's UK City of Culture Bid 2021 and was first presented at Warwick Business School in the Shard in London in June 2015. The original animation was made in Apple's Quartz Composer software, which reads image data from an attached camera and from which individual pixel data is used to stimulate the movement of the graphics as well as controlling the playback of audio loops in Ableton Live on a second computer. Subsequently, however, the generation of the graphics has been changed so that it is now performed in Processing 3. Inter-application communication is achieved using Open Sound Control. The system design is shown in figure 1.

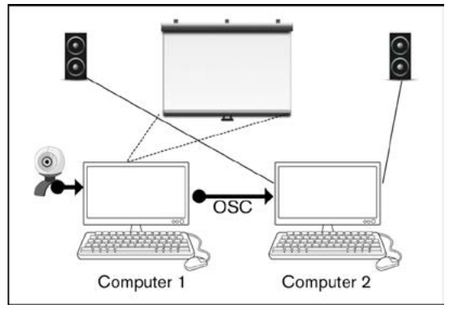


Figure 1. System design for *Oscilloscope*.[1] Computer 1 reads in image data from a USB webcam and reads pixel values using Apple's Quartz Composer software. Within Quartz Composer, these data increments the phase of sinusoidal functions, the outputs of which are sent via OSC to Processing 3, which generates the visuals to be projected. At the same time, when the outputs of these functions reach threshold values, trigger messages are sent via OSC to the second computer running Ableton Live to play or stop looped tracks.

This work, and in particular the interactive sound texture, draws influence from composers who have utilised principles of cybernetics, systems theory and complexity theory in their compositions, notably Brian Eno and Agostino Di Scipio. The work seeks to combine ideas utilised in Eno's generative music systems with Di Scipio's musical eco-systemic design. The ease with which these two technologically differing systems may be integrated is a testament to the shared cybernetic ontology that underpins the work of both composers. The focus of emphasis in the creation of this musical work is the cybernetic process, which significantly differs from usual approaches to computer music making.

2. CYBERNETICS AND ENVIRONMENTAL INTERACTION

Cybernetics is the science and study of systems and in particular how information flows between man, machine and environment in a matrix of feedback loops that may form emergent behaviours. While both composers have explicitly cited cybernetics as an influence on their work (Eno in [1], Di Scipio in [3]). They have also made explicit other composers who have utilised cybernetic techniques that have influenced their compositional process. Of particular interest to this paper, both composers build on compositional ideas espoused by Xenakis (Eno in [1], Di Scipio in [2] and [3]).

In 1963 Xenakis attempted to 'generalize the study of musical composition with the aid of stochastics' [4]. To this end he utilized the methodology found in W. Ross Ashby's 1956 book, *Introduction to Cybernetics* [5]. From this extrapolation of Ashby's work Xenakis further postulated that 'second order sonorities' would emerge from the interactions of sonic grains: the idea that the interactions of grains over time in the compositional process, at a 'micro level', would form timbres and compositional gestures at the 'macro level' (i.e. the grains, when combined in a certain way, would exhibit emergent behaviours). Xenakis first implemented his granular compositional technique in *Analogique A* (1958) for string ensemble and *Analogique B* (1958–59) for tape. Although both Eno and Di Scipio have criticised Xenakis' approach (Eno in [1], Di Scipio in [2]), the idea that emergent (musical) behaviour can arise from composed interactions underpins both composers' working methods.

At root both Eno and Di Scipio share the desire to create autonomous musical systems that are modelled on the way in which living systems generate complexity and that are also able to display emergent behaviour. Both composers reject the linear design ontology of the majority of interactive computer music systems in favour of ecosystemic systems design; a constructivist ethos in which the design of interactions of a system's components, prior to performance, takes precedence over a macro musical design, shaped by a composer in realtime during a performance. Di Scipio notes that, '[t]his is a substantial move from interactive music composing to composing musical interactions, and perhaps more precisely it should be described as a shift from creating wanted sounds via interactive means, towards creating wanted interactions having audible traces. In the latter case, one designs, implements and maintains a network of connected components whose emergent behaviour in sound one calls music' [3].

Eno first encountered cybernetics as an art student in Ipswich in the early 1960's under the tutelage of the telematic artist and cybernetics enthusiast Roy Ascott. Eno later read the cybernetician Stafford Beer's book *Brain of the Firm* (1972), from which he has extensively quoted and used as a justification for his compositional approach [6]. Eno States that: 'the phrase that probably crystallised it most [Eno's cybernetic approach to music]…says "instead of specifying in full detail; you specify it only somewhat, you then ride on the dynamics of the system in the direction you want it to go". That really became my idea of working method' [1]. Thus we may see a preoccupation with systemic deign in composition; one which is reliant on the setting of some initial parameters but equally relies on a medium which provides dynamic interaction. Pickering notes that, 'such systems can thematize for us and stage an ontology of becoming, which is what Eno's notion of riding the systems dynamics implies' [6]. Eno observes that this type of system generates 'a huge amount of material and experience from a very simple starting point' [7], further emphasising the cybernetic tropes of becoming and emergence.

Eno's generative music systems have been realised by a number of different technological means, including the VCS3 synthesiser, analogue tape manipulation and the KOAN generative music software. The method emulated in this composition takes inspiration from Eno's tape based composition *1/2*, from the album *Music for Airports* (1978). The design of *1/2* consists of individual vocal sounds (wordless "aaahhs" in the key of F Minor) recorded onto separate lengths of tape between fifty and seventy feet long [8]. To facilitate these long loops, the tape was spooled around metallic studio chair legs. Eno then recorded these non-contiguous loops back onto the multitrack tape: 'I just set all these loops running and let them configure in whichever way they wanted to.' [9] The complexity of the piece arises from the five-second vocal recordings, recorded on to tape loops of differing lengths, at times coalescing to form chords and shifting melodies and at other times leaving silence or only individual notes. The aesthetic effect is of a rather sparse angelic choir but producing a texture that is predetermined but not predictable. There is no meter or pulse but the notes appear to interact in a knowing and predestined way; the structure seems designed but at the same time beguiling.

The aesthetic effect of this piece demonstrates Eno's preoccupation with what Nyman called the 'cult of the beautiful' [10], but it also sees him engaging in the 'new determinacy' [10] techniques employed by his contemporary, English experimental composers, such as Gavin Bryars and Cornelius Cardew. However, Eno's version of the new determinacy is a strictly technological one, in which the timing and tone of the piece is mitigated by technological means. This is also a probabilistic process, but specifically designed to produce a class of goals. It is also noteworthy that the environment is active in the technological process. This is seen in the long tape loops, which are passed out from the tape recorder and spooled around objects such as metallic microphone stands and chair legs, the friction of which will alter the timing of each loop in a slightly unpredictable way.

Di Scipio's design ethos is one that encompasses the environment in the man/machine interaction, and thus embraces a tenet that is central to the cybernetic ontology. In fact he makes his cybernetic approach explicit when discussing interactive computer music: 'I try to answer (the question of interactivity) by adopting a system-theory view, more precisely a radical constructivistic view (von Glasersfeld 1999, Riegler 2000) as found in the cybernetics of living systems (Maturana and Varela 1980) as well as social systems and ecosystems (Morin 1977)' [3] With this paradigm in mind, Di Scipio approaches the question of interactivity from an ecological viewpoint: 'The very process of interaction is today rarely understood and implemented for what it seems to be in living organisms (either human or not, e.g. animal or social), namely as a by-product of lower level interdependencies among system components. In a different approach, a principal aim would be to create a dynamical system exhibiting an adaptive behaviour to the surrounding external conditions, and capable to interfere with the external conditions themselves [sic]' [3].

He further states that the system should be capable of being a 'self observing system' (independent from an agent/performer), one that is capable of tracking what happens both externally and internally and making adjustments accordingly. He sights Gordon Mumma's *Hornpipe* (1967) as a pioneering example of such a system [3]. Here, interaction is no longer agent acts, computer re-acts, as in the linear model; instead it becomes a fundamental structural element from which a system may emerge. The flow of energy in the system is no longer one way (i.e. from the composer in real time); energy may be derived from the environment and a composition may be self-sustaining, with little real-time input from a composer/performer. It becomes obvious that in such a system the design of the interactions between all the components are fundamental to the construction of the composition; without a considered, eco-systemic design, interactions will simply not occur. He states, 'I think that these interrelationships (between elements of a system) may, instead, be the object of design, and hence worked out creatively as a substantial part of the compositional process' [3].

Di Scipio is keen to assert that the vast majority of interactive computer music conforms to the aforementioned linear model and as such the eco-systemic, cybernetic approach reflects a "paradigm shift" in compositional approach [3].

3. SYSTEM IMPLEMENTATION

The emulation of Eno's tape-based system is achieved using the Ableton Live software. Loops of tape are substituted with non-contiguous loops of audio samples, which, when played simultaneously, never repeat the same sequence twice. Thus a complex, laminar and ephemeral compositional emerges. The sound materials that make up these loops reflect the aesthetics of the visuals. It must be stressed that in terms of this paper the resultant musical structures, while they may be considered aesthetically pleasing, are of secondary importance to the generative process through which they were constructed and how these are resultant of, and interact with, the environment and the visual material.

The shifting geometrical shapes in the visual material are made from two "rings", each constructed out of a triangle strip, joined together end to end. Figure 2 shows how the structure of one ring is made up with the shading removed and the lines of each triangle made visible. Alternate vertices of each ring's triangle strip define a closed loop and so the whole shape can be described by three loops, two at either end and a loop common to both rings at the shape's centre where the rings join. The shape, as seen in the animation, is shown in figure 3.

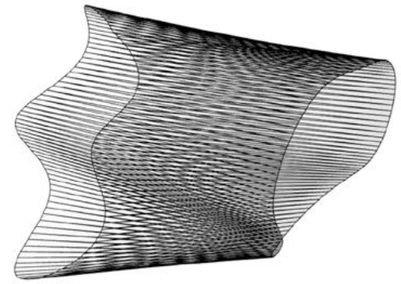


Figure 2. One of the rings that make up the moving shapes of the image. Lines connecting the vertices have been made visible to show how a ring is made from a triangle strip joined at both ends.

A closed loop such as a circle or ellipse is defined by a two-dimensional sinusoidal equation. Therefore, the animation of each of the loops can be achieved through modulation of the amplitudes and frequencies of their sinusoidal components. In figure 2 it can be seen that these equations determine the location of vertices in the *y* and *z* planes, but in addition a third sine wave component is used to vary the *x* positions of the vertices and thus modulate the "width" of the ring. In this work, these amplitudes and frequencies are themselves modulated by sine waves of fixed amplitudes but whose phase is incremented by pixel values from the image data obtained from the camera. Through this process, arbitrary motions can be created in response to the environment but within limits set by the creators of the work.

Complex motions can therefore be achieved with fairly simple mathematical processes similar to AM and FM synthesis processes familiar to the computer sound designer. Thus, although the basic shape is very simple, the layers of modulation processes on the shapes structure create complex bio-mimetic movement giving the visual effect of a highly-abstracted sea creature. However, the point of interest here is that this bio-mimesis arose, not from a 'top-down' design that seeks to emulate the totality of complex movements, but from the set up of multiple modulations whose unpredictable interactions generate patterns that can be seen as an emulation of the motion of a living organism. As a result of this mimesis, sound samples that have a direct correlation to water are used which reflect the emergent properties of the animation.

Further generative interactions were designed and inspired by Di Scipio's compositional method. Di Scipio's design ethos has been adhered to in the creation of this piece via the interactions between the environmental input of the camera source and the musical and visual software. A grid of twelve discrete point sources is derived from the incoming image produced by the camera and changes in light intensity of these sources indirectly trigger individual sample loops to play or stop. Intensity values increment the phase of sinusoidal functions of the

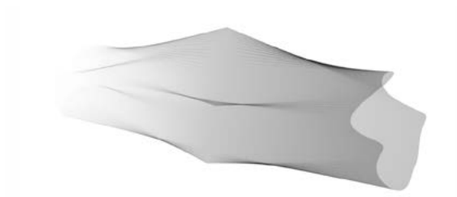


Figure 3. The complete shape (with connecting lines removed and shading added) made up of two rings.

visual material, which outputs “play” or “stop” messages to individual sequencer tracks once threshold values are crossed. In this way, the twelve light point sources are mapped to thirty sample loops to create a matrix of non-linear triggering possibilities. Thus, the rate of change of individual sonic and visual components within the installation are constantly changing in response to light conditions in the surroundings as read by the camera. The speed of one loop of the visual material also determines the tempo of the sequencing software so that a higher speed will generate more triggering opportunities. Many of the samples are also subject to real-time digital signal processing techniques which are controlled by automated control envelopes. The speed at which the envelopes move through their control cycle is determined by the tempo and thus changes with alterations in overall light intensity. Thus, through these structured interactions an autonomous autopoietic musical and visual system is achieved.

4. CONCLUSIONS

Although it is recognized that this installation is conceived in the digital domain, the title *Oscilloscope*—referring to a form of analogue computer display, was chosen to reflect the critique of common assumptions of digital technology that this work represents. The processing of discrete bits of information facilitates linear mapping formulae where a single input value produces a related output value. With such an approach, there is a tendency to produce complexity through accretion; either the accumulation of more inputs and outputs or the linear chaining of mappings between a single input or output. With this work, the aim was to avoid such linearity between the visual input data and the resulting material through the designing of low-level interactions between simple materials. The sounds and visuals of the piece are therefore not a mere sonification or visualization of input data but the result of processes driven by that data.

It is important to state that the system’s interactions are only indirectly implemented, as Di Scipio puts it, interactions are the ‘by-product of carefully planned-out interdependencies among system components, [which] would allow in their turn to establish the overall system dynamics, upon contact with the external conditions’ [3]. He also believes that this type of construction is akin to the mapping in living organisms that allows emergent behaviour to occur. The further coupling to the Eno system to Di Scipio’s increases the complexity of the interactions and further enhances the possibility of emergent musical behaviour.

5. REFERENCES

[1] D. Whittaker, *Stafford Beer, A Personal Memoir*. Wavestone Press, 2003.

[2] A. Di Scipio, “The Problem of 2nd-order Sonorities in Xenakis' Electroacoustic Music,” *Organised Sound*, vol. 2, no. 2, pp. 165–178, 1997.

[3] A. Di Scipio, “’Sound is the interface’: from interactive to ecosystemic signal processing,” *Organised Sound*, vol. 8, no. 3, pp. 269–277, 2003.

[4] I. Xenakis, “Musiques Formelles” Paris : Editions Richard-Masse 1963. Reference to the online version: http://www.iannis-xenakis.org/MF.htm Reference also to the American expanded edition Formalized music: Thought and mathematics in composition. Harmonologia series; no. 6. New York: Pendragon Press, 1992

[5] P.A. Kollias, “Music and Systems Thinking: Xenakis, Di Scipio and a Systemic Model of Symbolic Music,” *Proceedings of the Electroacoacoustic Music Studies Network International Conference*, Paris- INA-GRM et Université Paris-Sorbonne (MINT-OMF), 2008, pp. 213-218.

[6] A. Pickering, *The Cybernetic Brain, Sketches of Another Future*. University of Chicago Press, 2011.

[7] B. Eno. (1996). *Generative Music* [online]. Available: www.inmotionmagazine.com/eno1.html

[8] D. Sheppard, *On Some Faraway Beach, The Life and Times of Brian Eno*. Orion, 2008.

[9] G. O’Brien, “Eno at the Edge of Rock,” *Interview*, vol. 8, no. 6, pp. 269–277, 1978.

[10] M. Nyman, *Experimental Music, Cage and Beyond*. Cambridge University Press, 1999.

[11] E. von Glasersfeld. (1999). *The Roots of Constructivism* [online]. Available: www.oikos.org

[12] H. Maturana and F. Varela, *Autopoiesis. The Realization of the Living.* D. Reidel Publ., 1980

[13] E. Morin, *La méthode. La nature de la nature*. Seuil, 1997.

[14] A. Riegler, www.univie.ac.at/construtivism/

Continuous Order Polygonal Waveform Synthesis

Christoph Hohnerlein[†*], Maximilian Rest[†*], Julius O. Smith III[*]
[*]Center for Computer Research in Music and Acoustics, Stanford University, 660 Lomita Drive, Stanford, CA 94305, USA
[†]Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany
[chohner,mrest,jos]@ccrma.stanford.edu

ABSTRACT

A method of generating musical waveforms based on polygon traversal is introduced, which relies on sampling a variable polygon in polar space with a rotating phasor. Due to the steady angular velocity of the phasor, the generated waveform automatically exhibits constant pitch and complexly shaped amplitudes. The order and phase of the polygon can be freely adjusted in real-time, allowing for a wide range of harmonically rich timbres with modulation frequencies up to the FM range.

1. INTRODUCTION

Connections between geometric shapes and properties of associated sounds has long been an appealing field of interest for engineers and artists alike, ranging from the strictly physical visualizations of Chladni [1] to the text-based descriptions of Spectromorphology [2]. Highly complex patterns emerge from seemingly simply ideas and formulations, such as Lissajous figures [3] or phase space representations [4]. The relation between visual patterns, motion and sound has been both an inspiration and expression for decades [5, 6].

Vieira-Barbosa produced some excellent animations of polygonal wave generators [7]. While only working with integer order polygons, he also animated the concept of polygon phase modulation and produced interactive sonifications of the resulting waveforms.

Chapman extended this idea to arbitrary orders, but instead of sampling with a phasor into the time domain he uses direct geometric projection, resulting in sharp angular waveforms [8]. He also introduced a more rigid mathematical framework and uses the Schläfli symbol $\{p, q\}$ to denote the geometric properties of regular polygons as a ratio of integer values p and q [9].

Sampath provides a standalone application which allows the user to design a large set of waveforms from geometric generators [10]. Among others, these include Bezier curves, spirals, n-gons, fractals and Lissajous curves.

In the less graphical oriented domain, digital waveshaping synthesis by Le Brun might produce the most similar results to the synthesis method proposed here [11].

2. SYNTHESIS METHOD

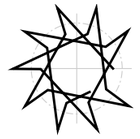


Figure 1: Example of a polygon which requires more than one cycle for a closed shape ($n = 3.33, T = 0.2$).

Polygonal waveform synthesis is based on sampling a closed-form polygon P of amplitude p with a rotating phasor $e^{j\phi}$. The fundamental pitch of the generated waveform is based on the angular velocity of the phase $\phi(t) = 2\pi ft = \omega t$ with sampling time t and fundamental frequency f. The polygonal expression $P(\phi, n, T, \Phi)$ simultaneously draws the polygon in the complex plane and generates the waveform when projected into the time-domain, as shown in Fig. 2.

2.1 Polygon

To create the polygon P, a corresponding order-dependent amplitude $p(\phi, n, T)$ is generated:

$$p(\phi, n, T) = \frac{\cos\left(\frac{\pi}{n}\right)}{\cos\left[\frac{2\pi}{n} \cdot \mathrm{mod}\left(\frac{\phi n}{2\pi}, 1\right) - \frac{\pi}{n} + T\right]}, \quad (1)$$

with the angle $\phi(t)$, order of the polygon n and a parameter T for offsetting the vertices, descriptively called *teeth*, adapted from [12].

Non-integer rational values of the order n require multiple cycles c of the phasor to yield a closed shape as depicted in Fig. 1. The number of cycles depends on the smallest common multiple between the decimal digits of the order and 1.

In Schläfli notation $\{a, b\}$, the rotations c corresponds to the second integer a. All polygons of the Schläfli symbol $\{a, b\}$ where $a > 2b$ may be produced. Then, the order is simply

$$n = \frac{a}{b}. \quad (2)$$

Furthermore, non-integer order polygons don’t necessarily need to close to avoid discontinuities, only the projection does. Figure 3 shows this for the bottom three waveforms.

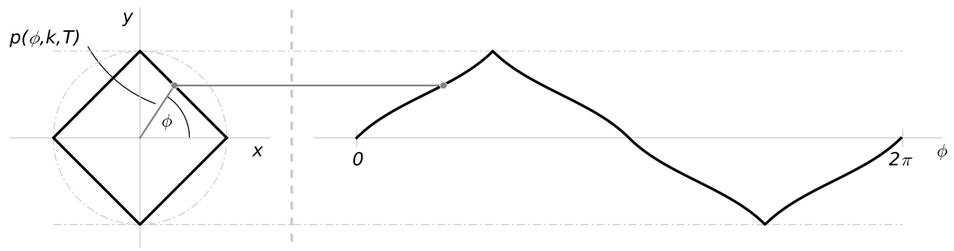


Figure 2: Projection of a square polygon (order $n = 4$) from the two-dimensional x/y plane into the time-domain.

2.2 Projection

The projection from the complex plane onto the time domain is done by simply taking the real (or imaginary) part of the polygon P:

$$P(\phi, n, T, \Phi) = p(\phi, n, T) \cdot e^{j(\phi+\Phi)} \quad (3)$$

$$x = \Re\{ P(\phi, n, T, \Phi) \} \quad (4)$$

$$y = \Im\{ P(\phi, n, T, \Phi) \} \quad (5)$$

Although Fig. 2 only depicts the extraction of the y component, it should be noted that the only difference to the x component is a phase shift by 90°. The additional phase offset Φ rotates the polygon in the complex plane and allows for phase modulation of the time domain signal.

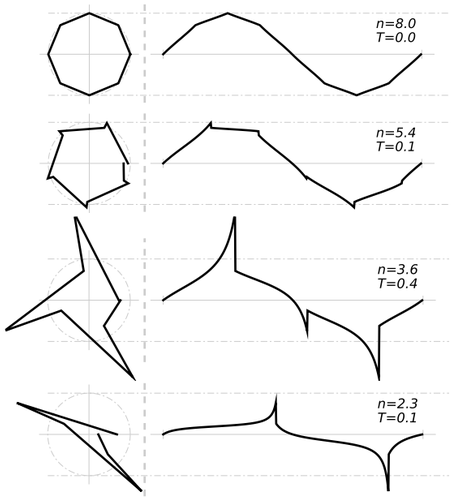


Figure 3: Projections of polygons $P(\phi, n, T, \Phi)$ from the 2D space into the time domain, $\Phi = 0$.

3. EVALUATION

In this section we discuss the three synthesis parameters order n, phase ϕ and teeth T and their influence on the sonic properties of the waveforms.

3.1 Order n

As the order $n \in]2, \infty]$ of the polygon is specifically not bound to be an integer, the shape of the polygon may be changed quasi-continuously in real-time. There is a hard lower limit of 2, corresponding to the polygon collapsing into a line, which, depending on the phase offset and projection, results in zero or infinite amplitude. For $n \to \infty$, the waveform approaches a pure sine wave. Figure 4 shows the spectrogram of a logarithmic sweep over the orders $n \in]2, 11]$ with a constant fundamental pitch of $f_0 = 100\,\text{Hz}$.

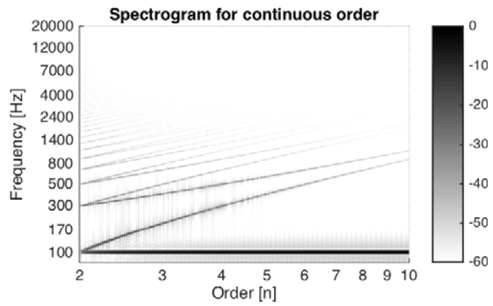


Figure 4: Spectrogram over order $n \in]2, 11]$, $f_0 = 100\,\text{Hz}$.

n	h_1	h_2	h_3	h_4	h_5	h_6
2.001	$1\,f$	$3\,f$	$3\,f$	$5\,f$	$5\,f$	$7\,f$
3	$2\,f$	$4\,f$	$5\,f$	$7\,f$	$8\,f$	$10\,f$
4	$3\,f$	$5\,f$	$7\,f$	$9\,f$	$11\,f$	$13\,f$
5	$4\,f$	$6\,f$	$9\,f$	$11\,f$	$14\,f$	$16\,f$
6	$5\,f$	$7\,f$	$11\,f$	$13\,f$	$17\,f$	$19\,f$

Table 1: Ratios of harmonic overtones to the fundamental f with increasing order. For non-integer orders, overtones are continuously interpolated.

At lower orders, strong harmonics form at specific ratios as noted in Table 1. They split and drift upwards with increasing order, until only the fundamental is left and the waveform is recognized as a pure sine wave.

3.2 Phase offset Φ

Modulating the phase of a polygon P by adjusting Φ is non-trivial for non-integer orders, as discussed in Section 2.1. For closed loop polygons, phase modulation results in interesting spectral behavior as shown in Figure 5, where the fundamental and even overtones are bend up while odd overtones are bend down.

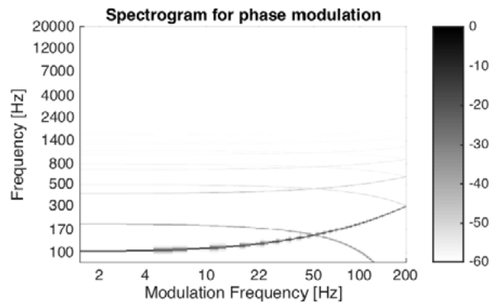


Figure 5: Spectrogram over phase modulation frequency, $f_{mod} \in [2, 200]$ Hz with order $n = 3$, $f_0 = 100\,\text{Hz}$. Visual artifacts due to non-continuities in sweep.

Depending on the speed of a continuous phase modulation, both slowly evolving shapes or harsh, metallic sounds may be generated.

3.3 Teeth T

The parameter T, named for its visual effect on the polygon, allows the over-extension of the polygon's vertices.

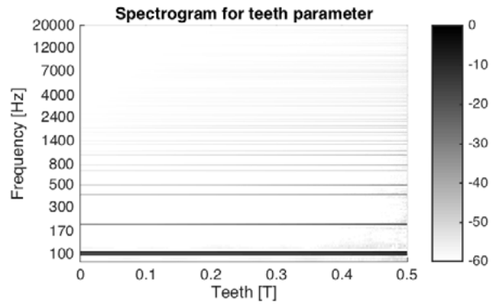


Figure 6: Spectrogram over the parameter Teeth $T \in [0, 0.5]$ with order $n = 3$, $f_0 = 100\,\text{Hz}$.

Increasing T can make the polygon exceed the unit circle and consequently overdrive the output amplitude. For lower values, this will only amplify present harmonics as shown in Figure 6, which shows the sonic effects of sweeping the parameter $T \in [0, 0.5]$. Depending on the employed limiting technique, higher values will drive the oscillator into saturation, allowing fine-grain control of additional harmonic partials.

4. IMPLEMENTATION

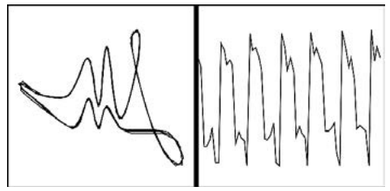
A monophonic version of the proposed synthesis method was implemented in Max/MSP [13] to explore the physical interaction with the available parameters. Figure 7 shows the GUI, with both the polygon and the time-domain signal drawn in real-time. Figure 8 shows two visually and sonically interesting polygons, their time-domain representation and their settings. [1]

[1] Please find a small selection of audio samples at `https://ccrma.stanford.edu/~chohner/polygon_samples.zip`

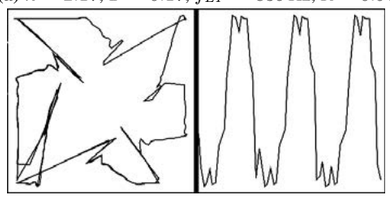
Most of the challenges when porting synth engines into a usable device, both virtual and hardware, are rooted in robustness and edge behavior. We highlight several aspects that need to be taken into account here:

4.1 (Anti)-Aliasing

One general artifact of digital synthesis is aliasing, see [14]. This specifically holds true for most of the waveforms generated here: they often contain discontinuities in the slope, which in turn result in high frequency content that is beyond the typical audio Nyquist limit of 22.05 kHz. To alleviate this, we propose generating the waveforms at four times the final audio sampling rate, lowpass filtering to the Nyquist limit using a 128th-order FIR, then decimating by a factor of 4. Such 4× oversampling dropped the aliasing below the noise level in our tests.



(a) $n = 2.17$, $T = 0.17$, $f_{LP} = 835\,\text{Hz}$, R = 0.94



(b) $n = 5.72$, $T = 0.91$, $f_{LP} = 835\,\text{Hz}$, R = 94

Figure 8: Polygons and corresponding waveforms as visualized by the proof-of-concept implementation.

4.2 Lookup Table vs. Phase Accumulation

Digital waveforms can be either generated as a lookup table or on the fly, or in a mixed approach. Lookup tables might generally be faster and can be pre-antialiased, but in this case require a sophisticated layout or interpolation to accommodate the various lengths of the waveforms due to the required cycles c to close a non-integer shape. In the prototype we chose to evaluate Equation (5) with a continuously varying angle ϕ, accepting an increase of high frequency noise when rotating non-integer orders.

4.3 Amplitude Limiting

As mentioned in Section 3.3, non-zero values of T result in amplitudes that can exceed the unit circle. To keep the waveforms in arbitrary but strict amplitude limits, clipping or compression must be applied to the output signals. A simple hard clipper with a variable input attenuator is applied to the oversampled signal in our implementation to keep the audio signals within [-1,+1] limits.

4.4 Filtering

A traditional sculpting lowpass filter as known from subtractive synthesis is employed to further shape the pro-

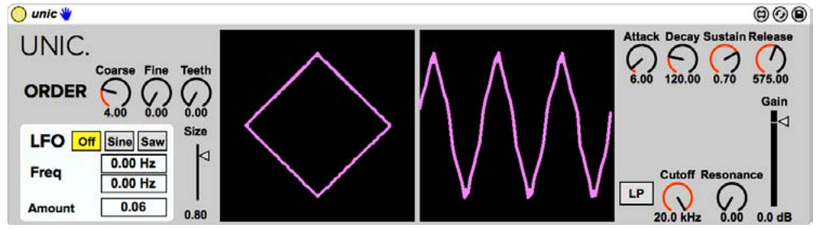


Figure 7: Screenshot of the proof-of-concept device in Max-For-Live.

duced waveforms. As expected, this introduces rounded vertices in the polar domain and overshoot depending on the resonance setting.

Any polygon with a geometric centroid different from 0 additionally introduces a dc offset when projected into the time domain. While this can be done deliberately to produce modulation signals, it should be avoided in the audio domain. A DC blocker [15] is implemented at the output of the synth.

4.5 Phase Modulation

A simple phase modulation scheme based on an LFO was implemented, which can be toggled between sinusoidal and linear waveforms. Very slow LFO frequencies allow for ever changing soundscapes, whereas modulation frequencies in the audible range allow for FM-esque sounds [16].

5. CONCLUSIONS

The proposed continuous-order polygonal-waveform synthesis is able to generate a wide variety of timbres, ranging from more traditional waveforms such as square and triangle to harsh digital sounds. The unusual control parameters give a new approach to modulation and our test implementation shows that interacting with it is quite rewarding.

Future work should include the effect of adding additional voices which can be synced, detuned and cross modulated. LFO tracking and more sophisticated filter topologies would also open further sonic sculpting capabilities. Because of its immediate visual appeal, an implementation with a larger display surface should be employed, which could happen both in the virtual as well as the real world.

Acknowledgments

The authors would like to thank CCRMA for all the opportunities during their research stay and Jens Ahrens for the great support.

6. REFERENCES

[1] M. D. Waller and E. F. F. Chladni, *Chladni figures: A study in symmetry*. G. Bell, 1961.

[2] D. Smalley, "Spectromorphology: explaining soundshapes," *Organised sound*, vol. 2, no. 02, pp. 107–126, Aug. 1997.

[3] J. A. Lissajous, *Mémoire sur l'étude optique des mouvements vibratoires*, 1857.

[4] D. Gerhard and others, "Audio visualization in phase space," in *Bridges: Mathematical Connections in Art, Music and Science*, 1999, pp. 137–144.

[5] J. Whitney, *Digital Harmony: On the Complementarity of Music and Visual Art*. Byte Books, 1980.

[6] B. Alves, "Digital Harmony of Sound and Light," *Computer Music Journal*, vol. 29, no. 4, pp. 45–54, 2005.

[7] L. Vieira-Barbosa. (2013 (accessed Dec 1, 2015)) Polygonal sine animation. [Online]. Available: http://1ucasvb.tumblr.com/post/42881722643/the-familiar-trigonometric-functions-can-be

[8] D. Chapman and M. Grierson, "N-gon Waves–Audio Applications of the Geometry of Regular Polygons in the Time Domain," 2014.

[9] H. S. M. Coxeter, *Regular polytopes*, 3rd ed. New York: Dover Publications, 1973.

[10] J. Sampath. (2016, Feb.) DIN Is Noise. [Online]. Available: http://dinisnoise.org/

[11] M. Le Brun, "Digital waveshaping synthesis," *Journal of the Audio Engineering Society*, vol. 27, no. 4, pp. 250–266, 1979.

[12] User: Raskolnikov. (2011, May) Parametric equation for regular n-gon. [Online]. Available: http://math.stackexchange.com/questions/41940/is-there-an-equation-to-describe-regular-polygons

[13] M. Puckette, D. Zicarelli *et al.*, "Max/Msp," *Cycling*, vol. 74, pp. 1990–2006, 1990.

[14] T. Stilson and J. Smith, "Alias-free digital synthesis of classic analog waveforms," in *Proc. International Computer Music Conference*, 1996.

[15] J. O. Smith, *Introduction to Digital Filters with Audio Applications*. W3K Publishing, 2007, https://ccrma.stanford.edu/~jos/filters/.

[16] J. M. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the Audio Engineering Society*, vol. 21, no. 7, pp. 526–534, 1973.

Textual and Sonic Feedback Loops: Simultaneous conversations as a collaborative process in cmetq

Christopher Jette
Independent Artist
christopherjette@gmail.com

Nathan Krueger
University of Wisconsin - Oshkosh
kruegern@uwosh.edu

ABSTRACT

cmetq is an concert length work for baritone voice with live processing, fixed electronics and video projection. The text was created to highlight notions of etiquette associated with the emergence of the telephone in the 19th century and social media/mobile telephony in the 21st century. The text was collaboratively realized and began as a series of tweets between the authors that were collected and edited. This paper will articulate the motivation that influenced the formal design as well as the unique workflow for composing cmetq. This collective development of a concert length work results in a synergy, exploiting the unique assets of the constituent collaborators.

Tags: Composition Systems and Techniques, Collaborative Work

1. INTRODUCTION

cmetq is a concert length stage work for baritone voice with fixed electronics and live processing. The work collects statements around notions of communication etiquette as it relates to the 19th century telephone and 21st century cell phone and social media. The title of the work *cmetq* (pronounced c m e t q) is a compression of the words communication etiquette. This compression is a nod to the Hungarian Notation system and the title, collaborative methodology and dramatic scope of *HPSCHD* by John Cage and Lejaren Hiller. This compression reflects the intermingling of automated and intuitive processes that were used to create *cmetq*.

The work focuses on the language of etiquette surrounding communication technology. This paper documents the collaborative methods used to generate both the text and the musical composition. The text of *cmetq* was culled from a conversation between the authors, conducted over social media. The compositional and melodic material was created via an exchange between the performer and composer using sound recordings. We supplemented these exchanges with telephone, video conferencing, email, and in-person conversation. McLuhan reminds us "The medium is the message" [1] and *cmetq* emphasizes this influence of the medium on the message. The working process was designed to reveal the effects and artifacts of the medium. In compiling the final work these effects, such as brevity, the character of language and the quality of sonic material, define the complexion of *cmetq*. The creation of *cmetq* was dominated by two conversations and supplemented with several side channels of conversation. This paper will illustrate the two main conversations, the development of text and musical material, and the contributions of each process to the realization of *cmetq*.

We begin with a look at the conceptual framework for the *cmetq* project, discussing the relationship between our subject, communication etiquette, and the resulting composition. The following section investigates how the feedback loop of conversation serves as a model for the collaborative strategies. The conversation that generated the text occurred predominantly on social media, erupting in short bursts and moving across material in a nonconsecutive fluid manner. In contrast, the creation of each song follows a sequence from text to complete song, occurring as a series of recordings. Finally the composer reports about a compositional translation tool and how it evolved as a result of this particular work.

2. CONCEPTUAL FRAMEWORK

> Once the world is technologized, we can not go back. - *Nicholas Carr* [2]

cmetq is designed to motivate the listener to consider the unique position of technologies in our daily life. This is inspired by others who've asked similar questions, such as Youngblood [3] in *Expanded Cinema*.

> What happens to our definition of "family" when the intermedia network brings the behavior of the world into our home, and when we can be anywhere in the world in a few hours?

The relationship to technologies is manifest in the behaviors that arise socially. With the growth of population and the increasing range of inter-human communication options, the social aspect of the human experience expands. In the guise of etiquette, we collectively agree to a set of interpersonal rules. We vote on these rules through our actions, endorsing with adherence to convention and challenging by ignoring conventions of behavior. *Cmetq* is not a lexicon of the rules of etiquette, but rather a collection of observations that are drawn from considering the conversation around etiquette in both the present and historical realms. The authors are asking a question about the evolution, or lack thereof, in social conventions.

To generate text for the work we sent articles and comments to each other. From this conversation, we extracted particularly compelling lines that served as the text for songs. Presented as contiguous songs, these statements create a mosaic that invites the audience to discover connections, consider the unique social pressures of the era(s), and reflect on continuity of social trends across temporal divisions. In an era where social media provides a continuous flow of minutia, we endeavor to mimic the surface banality of a social media flood and position our chosen text as a unique filter. Where commercial products sift through data with an algorithm that is tethered to an advertising budget, the text of *cmetq* is adapted from the perspective of both message and sonic result. This approach leverages the caprice and artistic perspective of the collaborators and reflects the conversation occurring around these ideas.

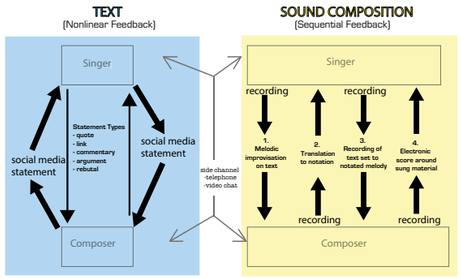


Figure 1. Parrallel Feedback Loops.

The creation of the work embodies how technical mediation is changing the essence of conversation. Our collaborative process embeds an embodiment of two different types of conversations in the design process of the work, the textual discussion that results in words for songs, and the sonic exchanges that results in the melodic and accompanying material (see Figure 1). Occurring simultaneously over fourteen months, these two approaches each leverage the characteristics of the medium in which they occur.

The first of these exchanges was the development of the text (see TEXT in Figure 1). Posting quotes, links, and commentary, the development of the text for *cmetq* occurred largely in a realm of social media. The textual restriction of twitter resulted in a conversation of short phrases. We refined the text in order to emphasize this brevity, shaping the text structures to represent the brevity and whimsy of internet memes or sound bites.

The second of these conversational feedback loops occurred as a sequential series of audio transmissions (see SOUND COMPOSITION in Figure 1), mirroring the sonic bias of telephony and the leisurely pace of letter writing. Developed through a series of recordings, the composition follows a linear pattern with the development of each song. The decelerated pace of this process invited time for consideration, reflection, and development of material, emphasizing craft in creating each iteration of the song.

Moving past the development phase to the presentation phase, *cmetq* is a stage work combining sonic and visual material. For this aspect of the work the composer generated some procedural drawing videos in the *Processing* language. These were in turn given to the visual artist Alejandro Casazi who, like the composer using the recording of vocal improvisations, utilized these videos as raw material for a visual narrative that frames and underscores the work (see Figure 3). The video's abstract imagery functions as a setting for the action onstage. To help intensify the distinction between the three characters, a unique color is utilized for each. To provide continuity with the score, the imagery is organized in a manner that reflects the texture and pacing of the electronic accompaniment.

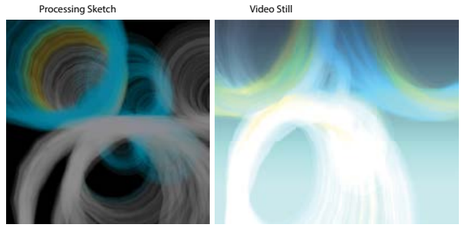


Figure 2. A still from the composer on the left and a still from the final video.

3. COLLABORATIVE METHODOLOGY

The collaborative methodology used to create both the text and the musical composition reflects how technology has altered the etiquette of communication. Here we discuss the specific steps involved in the development of the text and the composition. The various contributions of the composer and the performer are illustrated in Figure 3.

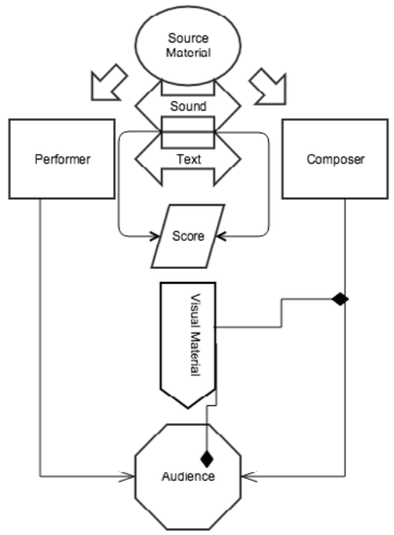


Figure 3. Flow from conception to presentation.

3.1 Generating text

The foundation for most vocal compositions are the texts that they are based upon. A significant hurdle to overcome in this project was the text source. Rather than choosing an existing text to set to music, the collaborators decided to create text based on communication technology and etiquette by having a conversation on social media. The conceptual basis of the piece originated as a playful quip regarding cell phones. This evolved into a larger conversation around etiquette and the idea of mining text from a social media conversation. This format provides several advantages. First, the imposed brevity allows the flexibility of maintaining a conversation over a long temporal frame with the immediacy of constant updates. Secondly, it presents the opportunity to generate a text that reflects our tastes and sensibilities. Finally, it allows the authors to explore a creative or productive use for social media channels and to utilize the persistence of the cyber-footprint as a bibliographic trail.

The work proceeded as a conversation on twitter and tumblr where we would first share an article or other primary source and then tweet our reactions to the material. From this corpus of largely personal interaction we extracted the most salient and interesting material, defining the aesthetic of the text component with our taste. As Kenneth Goldsmith [4] puts it his introduction to *Against Expression*

> If you can filter through the mass of information and pass it on as an arbiter to others, you gain an enormous amount of cultural capital. Filtering is taste.

To further reinforce our aesthetic position this material was then edited and shaped throughout the translation process. There were several ways in which this process happened. In some cases, phrases that resonated with the authors were extracted and edited to be retold in our own unique voice. Other times reactions were distilled into a single phrase that summarizes an entire article or an aspect of it. The performer then selected portions of text from the social media forums, edited as necessary, and then improvised melodies on the text.

The final iterations reflect our ideas around etiquette and create space for the audience to both have opinions and interpretations. In the end, thirty-eight statements were chosen from the larger conversation. At this juncture, the authors considered the narrative framework of the text. In the end, we conceptualized a quasi-narrative where the statements would be spoken from two distinct points of view. Since the text juxtaposes both a contemporary and historical dialog we created two characters, both portrayed by a single performer. These juxtaposed characters represent the unique intersection of wall mounted telephones and cell phones that defines a very particular temporal moment. Exploring the similarities across epochs, these characters exhibit the similar reactions to different forms of technology that occur in their respective era. One character lives in the epoch where the first telephone was invented, and the second character lives amidst the advent of mobile communication and social media. The audience is located in a present that for the moment knows both worlds.

We began the process of dividing the statements by having a conversation about the meaning of each statement and assigning a subtext. The arrangement of the text suggests a flow from one mental state of being to another, reflecting the typical mental process of a human brain, moving from one idea/concept to another. This led to what became the final order of the text, an emotional journey that examines a wide gamut of emotions and reactions to etiquette and technology. In the end, we used only twenty-five of the thirty-eight statements.

Throughout this process, it became clear that we needed to devise a way to infuse the piece with our own commentary. In order to not distort the two-character structure we created a narrator component. The text for this narrator is a series of soliloquies. They not only provided context for the characters, but they also enabled us as the creators of the piece to comment directly in declarative statements. For example, a soliloquy in the third act stresses the role of creativity in how we use communication technology.

> In tracing the tales of communication that span one century the shift in what defines the topography of etiquette reveals that our potential is bound by the grasp of our aesthetic imagination.

3.2 Compositional Process

The composition of *cmetq* is rooted in a collaboration between composer and performer. The translations of voice via software and extensions via digital sound opens new opportunities, as Risset [5] notes

> digital sound should be used to expand the sonic world, as Varese longed to do, to take advantage of our perceptual features, to explore new territories, and to invoke powers of the inner self.

The compositional mechanism that enables this approach is a translation of recordings into notation, extending the composers previous work. [6] In *cmetq*, the motivation is similar to this earlier work but the actual translation process has been modified. The motivation for this translation technique is to utilize not only the sonic material of the performer involved but also the unique performance aspects that a performer brings to their instrument. In the case of the voice, the instrument is highly personal and the unique spectral profile and morphology of the performer is a rich point of departure. From the performer's perspective, the opportunity to provide musical material is uncommon and artistically stimulating.

3.2.1 Translation Technique

In previous iterations, the translation process located functionality of the procedure in different programs, such as analysis in one program and editing in another. In order to create a single melody, four to eight different realizations were edited into a single phrase by hand. Where previous projects had up to ten melodies, *cmetq* had thirty-eight textual phrases to be translated. To improve the efficiency of the process, a different approach is utilized. Instead of manually editing together multiple takes, real time controls were added and the composer rehearses and then performs the translation in a single take. By incorporating realtime performance, the system becomes more efficient. This change is implemented in two ways, rendering a score in real time and by dynamically controlling the lag time of the autocorrelation pitch estimation.

In this implementation we use an autocorrelation pitch follower implemented in *SuperColliders* `Pitch` *UGen*. As noted by Roads, autocorrelation is most efficient at mid and low frequencies. Thus it has been popular in speech recognition applications where the pitch range is limited. [7]. Working with vocal material of a relatively short length, autocorrelation was able to resolve the pitch content of the singer.

The first means of providing immediate feedback is the generation of a realtime score. To accomplish this the *bach library* [8] in *MaxMSP* environment is employed. Using *OpenSoundControl* the midi note value of the detected pitch is sent from *SuperCollider* to the *MaxMSP* environment where the `bach.transcribe` object is utilized to format the incoming information and present it via a `bach.roll`. This immediate presentation enables the composer to quickly judge the accuracy and usefulness of the translation and if need be, alter the parameters of `Pitch` *UGen*. To further judge the effectiveness, the transcription can be played back with a simple midi instrument while simultaneously playing the audio source. If the translation is judged suitable the bach library enables the quantization of the `bach.roll` into `bach.score` object. Having both the raw spacial notation and a quantized version side by side for both visual and auditory review means the optimal translation can be quickly determined with a few alterations of quantization settings. Once quantized the information is output as a `musicxml` file which is brought into *Finale* and the text is set.

The other control that was utilized in the rendering translations was the dynamic control of the rate at which the pitch analysis is performed in *SuperCollider*. The analysis routine utilizes a trigger for the rate at which pitches are reported. In previous versions of the translation process it was optimal to set the trigger to rapidly report notes. This not only renders all of the slight variations in pitch but also helps to show more precisely where a change in pitch occurs. The downside to this approach is that there is an excess of information that the composer must reduce. The addition of dynamic control means that through focused listening and several rehearsals, the composer can control the reporting rate to approximate the ideal rate per each section of the sound file.

3.2.2 Text Setting

Once these translations were completed, they were sent to the performer. The performer took the original text and reset it, making minor edits to melody, rhythm, and text as needed. In some cases, rhythms were adjusted for purpose of syllable stress and syllabification. In other cases, certain words in the phrase were extended to become melismatic, which supported the original integrity of the translation. These reworked melodies were recorded and sent to the composer.

The recorded melodies serve as a sonic point of departure for the composer in creating the final works. The melodies are set with a fixed electronic accompaniment. The goal of these settings is to create a series of unique songs that explore the ideas that the collaborators discussed with each text. Each melodic phrase was approached differently, often using excess material from the initial translations and aims to create songs which can stand on their own and work in the larger piece. The final compositional stage was the creation of connective sonic material between the successive songs. This material took the form of brief fixed electronic works.

4. CONCLUSIONS

The creation of *cmetq* was motivated by the authors' interest in etiquette and its relationship to technology. It is based to two simultaneous conversations. First, the discussion around the development of the text which explores etiquette and communication technology, while using various social media channels to maintain that conversation. Second is the development of the sonic material with the composer and performer communicating through recordings. Each of the conversations is supplemented with conversations via telephone and video chat. The formal design and workflow of *cmetq* were directly influenced by both conversations and results in a unique performance piece. It is through conversation, in multiple modalities that we discovered the optimal form of the piece and how to ideally articulate ideas in sound.

Acknowledgments

A significant and heroic effort was put forward by Alejandro Casazi in realizing the visual aspects of this work. *cmetq* is able to function on stage as a result of his brilliant work. We also indebted to the Grant Wood Art Colony and the University of Wisconsin Oshkosh Music Department for their support of this work.

5. BIBLIOGRAPHY

[1] M. McLuhan, *Understanding Media: The Extensions of Man*, , McGraw-Hill, New York, NY (1964).

[2] N. Carr, *The Shallows: What the Internet Is Doing to Our Brains*, W.W. Norton, New York, NY (2010).

[3] G. Youngblood and R.B. Fuller, *Expanded Cinema*, P. Dutton & Company (1970). pp. 52

[4] K. Goldsmith and C Dworkin, Editors, *"Why Conceptual Writing? Why Now?" Against Expression An Anthology of Conceptual Writing : An Anthology of Conceptual Writing*, Northwestern University Press, Evanston, IL (2011).

[5] J.C. Risset, *Sound and Music Computing Meets Philosophy*, Proceedings of the 2014 International Computer Music Conference, (2014).

[6] C. Jette and K. Thomas and J. Villegas and A. Forbes, *Translation as Technique: Collaboratively Creating an Electro-Acoustic Composition for Saxophone and Live Video Projection* , Proceedings of the 2014 International Computer Music Conference, (2014).

[7] C. Roads, *The Computer Music Tutorial*, MIT Press, (1996).

[8] A. Agostini and D. Ghisi, *A Max Library for Musical Notation and Computer-Aided Composition*, Computer Music Journal 39 (2015). pp. 11-27

Tectonic: a Networked, Generative and Interactive, Conducting Environment for iPad

Lindsay Vickery
Edith Cowan University
l.vickery@ecu.edu.au

Stuart James
Edith Cowan University
s.james@ecu.edu.au

ABSTRACT

This paper describes the concepts, implementation and context of Tectonic: Rodinia*, for four realtime composer-conductors and ensemble. In this work, an addition to the repertoire of the Decibel Scoreplayer, iPads are networked together using the bonjour protocol to manage connectivity over the network. Unlike previous Scoreplayer works, Rodinia combines "conductor view" control interfaces, "performer view" notation interfaces and an "audience view" overview interface, separately identified by manual connection and yet mutually interactive. Notation is communicated to an ensemble via scores independently generated in realtime in each "performer view" and amalgamated schematically in the "audience view" interface. Interaction in the work is enacted through a collision avoidant algorithm that modifies the choices of each conductor by deflecting the streams of notation according to evaluation of their "Mass" and proximity to other streams, reflecting the concept of shifting Tectonic plates that crush and reform each other's placement.*

1. INTRODUCTION

TECTONIC: Rodinia is a work for four realtime composer-conductors and ensemble. In geology Rodinia is the name of a supercontinent that contained most of Earth's landmass between 1.1 billion and 750 million years ago. Tectonic can mean both 'the study of the earth's structural features' and 'the art of construction' and this works reflects both aspects of the word's meaning. The concept of slowly shifting plates that crush and reform each other's placement is the central paradigm of the work.

Rodinia is the second in a series that began with *Tectonic: Vaalbara* [2008]. In *Vaalbara* five instrumental streams are performed independently, using computer generated metronome pulses to manipulate the tempo of each stream, allowing the blocks of musical material to slide, grate and collide with one another like tectonic plates.

In *Rodinia* four composer/conductors control separate streams of graphical notation and audio (comprising live instruments reading the notation and their processed audio components) that interact through the algorithmically evaluated Mass and proximity of each stream. The work is performed using the Decibel Scoreplayer on multiple iPads via a manually connected network allowing for each participant conductor or performer to identify independently on the network [1]. The manually connected network was first used in Laura Lowthers' work for the Decibel ensemble, *Loaded* [2015]. Previous scores had prioritized synchronization between multiple iPads in order to present uniform representation of fixed scores for all performers. It is made possible by the adoption of the bonjour protocol to manage connectivity over the network. The use of the bonjour protocol also allows connectivity via OSC to stream data to other devices. In *Rodinia* this is used to stream generative data to a dedicated computer using Wave Terrain synthesis to process and spatialise the audio from the ensemble.

2. IMPLEMENTATION

Rodinia employs generative scores for each of the four streams directed by the composer-conductors. Unlike previous generative notation works by Vickery such as *Lyrebird* [2] and *The Semantics of Redaction* [3] *Rodinia* does not use the analysis of a pre-existing audio artifact to generate notation.

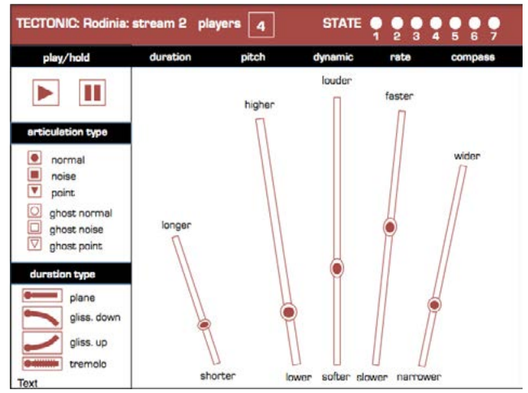


Figure 1. *Rodinia* conductor controller interface

Each composer/conductor in *Rodinia* uses an iPad interface, the "Conductor View", to generate notation for their group (Fig 1.). The controller interface is operated by two hands (the iPad permits 11 simultaneous multi-touch points) [4] allowing parameters to be specified simultaneously by the Left hand (play/hold, articulation, duration type) and Right hand (duration, pitch, dynamic, rate and compass). The variables Conductor View interface are:

- Players – defines the number of performers in each stream and generates a part of varied shade for each performer;
- State – saves a particular configuration of parameters that can be accessed at a later point;
- Play/Hold - stops and starts the generation of new notation;
- Articulation type – defines the graphical shape of the notation events;
- Duration type – generates alters the morphology of the notation events (line, curve up/down and tremolo);
- Duration – generates events of statistically longer or shorter duration;
- Pitch – designates the central pitch of the notation;
- Dynamic – generates larger/louder or smaller/softer notation events; and
- Compass – designates the statistical range that notation events fall within.

These parameters define the boundaries of stochastically generated graphical events which are distributed to the all of the iPads belonging to the same stream on the network. Like many works for the Decibel Scoreplayer, the notation for the performers is scrolled right to left across the iPad screen: in *Rodinia* this is designated the "Performer View" (Fig. 2). The scroll time, the duration between the notation's appearance on the right of the screen and its arrival at the "playhead", is 12 seconds. The playhead is - a black line of the left of the screen at which the performer's execute the notation [5]. This produces a scroll-rate of between 1.1 and 1.8 cm/s depending on the iPad model, falling below the maximal eye-hand span of the average sight-reader (less than 1.9 cm) [6][7]. Therefore, the musicians do not perform the notational event until it arrives – 12 seconds after specification by the conductor. This allows for the performers to comfortably "look ahead" at on-coming notation and for the conductors to evaluate strategies to avoid (or seek) collision with the other 3 streams.

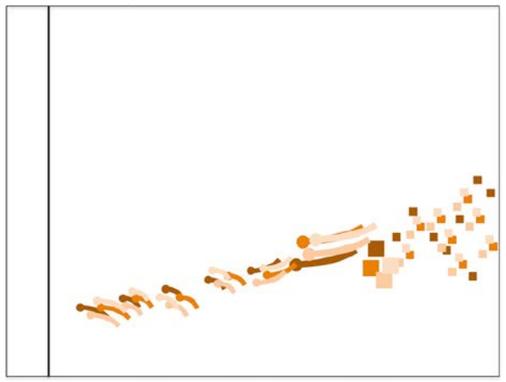


Figure 2. *Rodinia* Scoreplayer "Performer view" of Stream 1.

Rodinia also amalgamates the notation from each stream into a single score, the "Audience View", to be shown on a large screen behind the performers for both the audience and the conductors. Unlike the performer view, audience view shows the streams of notation approaching from four directions (left, right, top and bottom) (Fig. 3). The notation "wraps" around each time it completes the crossing from one side of the score to the other. As notation does not appear until the moment at which it is executed by the performers, the audience see it at the moment that it is heard.

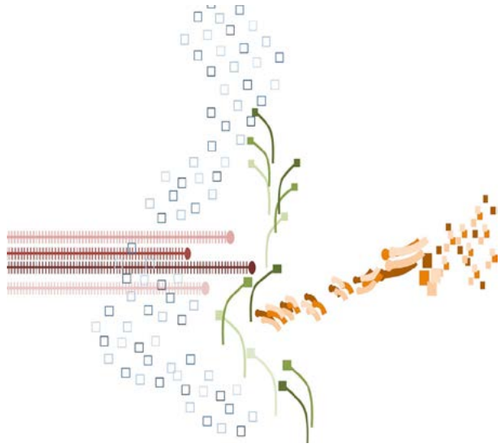


Figure 3. *Rodinia* Scoreplayer "Audience view"

The use of an audience view was first employed for the Decibel Scoreplayer in Vickery's work with Jon Rose *Ubahn c. 1985: the Rosenberg Variations* [2012]. For this and other rhizomatic works [8] the projected Audience View provides an overview of the current position of each player and graphically illuminates the choices taken in each stream.

Rodinia employs a collision avoidant algorithm which may modify the choices of each conductor. As notational streams approach one another they are pushed upward or downward according to their evaluated mass. Mass is defined as the density (duration, dynamic and compass) multiplied by the weight (articulation type and proximity) of each stream. Notation streams with a higher force deflect those of a lower force proportionally, spatially higher streams deflect upwards and lower stream downwards, if the streams are of equal height and mass the direction of their deflection is chosen randomly (Fig. 4).

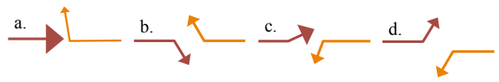


Figure 4. Collision avoidant using force evaluation: a. strong(L)/weak(R) interaction, b. weak(L)/weak(R) interaction, c. medium(L)/weak(L) interaction, and d. spatially higher stream deflects lower stream downwards.

This approach is similar to that adopted in Chappell's "self-avoiding" curve drawings [9], and Greenfield's "Avoidance Drawings" [10]. Chappell describes his process in the following way:

> *To generate a self-avoiding curve, I place "antennae" on the moving point that sense when the path is about to be crossed. . . . If the left antenna crosses the path, then the point executes a 180◦ reversing turn to the right* [11].

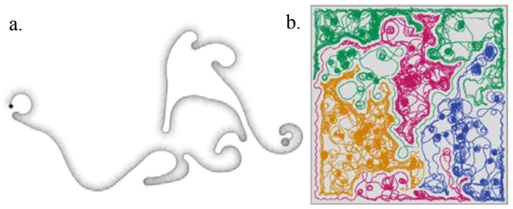


Figure 5. a. example of a point in the plane performing a self-avoiding random walk using Chappell's model. b. Greenfield's "avoidance drawing" (2015).

The key difference in *Rodinia* is that since music is a time-based medium, it can never "double-back" on itself and therefore in a generative score the deflection can never be greater than 90º.

Early studies conducted in Jitter, by Vickery for testing "collision avoidant lines" explored this paradigm, exploring "proximity only" avoidance (all lines were of equal density) to illustrate the kinds of pathways generated by this strategy (Fig. 6).

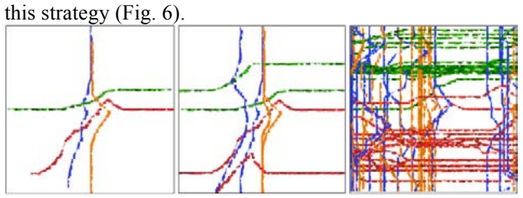


Figure 6. Vickery "collision avoidant lines" study for *Tectonic: Rodinia* (2013): first, second, and twelfth passes.

In Rodinia, a mass is calculated for each stream, M_S, based on its *cumulative density*: that is, based on the positions of the right-hand parameter sliders selected in the conductor view. This is based on both horizontal and vertical density as pictured in the score view.

The deflection angle of each stream, θ, is based both on the current mass of each stream calculated individually, as well as the total mass. If the distance between the leading point of each stream is below 175px the deflection angle rises from 0º to 90º exponentially in inverse of the proximity, as the proximity approaches 0px, such that:

$$\theta = \frac{\beta M_S \theta_D}{M_T} \qquad (1)$$

where θ is the new angle calculated individually for each stream, M_S is the mass of the same stream, M_T is the total mass, θ_D is the angle scalar, and β is a positive or negative scalar determining a turn in direction either left or right of the current direction of each stream. The height parameter is used to calculate whether an interaction results in an upward or downward deflection. The total mass, M_T, is the sum of all stream masses such that:

$$M_T = M_A + M_B + M_C + M_D \qquad (2)$$

3. NOTATIONAL CONVENTIONS

The notational paradigm, semantic spatial notation, employed by *Rodinia* has been developed over a number of projects by composers working with the Decibel Scoreplayer - in particular the approach to presenting notational events used in the generation of scores from John Cage's *Variations I* and *II* by Decibel [12] Fig. 7.

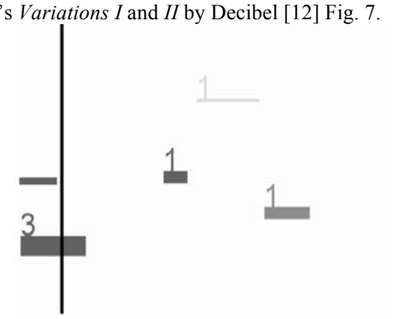


Figure 7. Decibel's scrolling, proportionally notated screen-score for Cage's *Variations I*.

The notation draws on conventions established in works by Cage and his colleagues Earle Brown and Christian Wolfe [13], chiefly proportional notation in which the vertical height of the notational event signifies relative pitch (relative to the range of the instrument), horizontal length its (absolute) duration and thickness its dynamic.

Unlike Decibel's scores for *Variations I* and *II*, in *Rodinia* timbre is indicated by the shape of the notational event rather than the shade. Performers are expected to match the qualities of timbral notational types (such as "normal" tone (rich harmonic sounds), "ghost" tone (harmonically poor sounds) and "noise" tone (inharmonic dense sounds)) within each stream. Each conductor controls a group of instruments of similar range so that register choices by the conductors are mirrored in the ensemble.

The streams, and individual parts within a stream are differentiated using shades of four principal colours orange, red, green and blue. Green-Armytage claims that 26 colours should "be regarded as a provisional limit – the largest number of different colours that can be used before colour coding breaks down" [14]. *Rodinia* is conceived for an ensemble of 16 performers (4 per stream) falling within the limits that of colour differentiation.

4. AUDIO PROCESSING APPROACH

The audio of the live instrumentalists is captured and processed digitally in Max/MSP on a standalone computer that is also networked via the bonjour protocol with the

iPad scores. This processing is informed by the movements of the four user controlled streams in order to generate and gradually deform a two-dimensional terrain map [15].

The terrain is initially generated by a method of perlin noise functions and undergoes both spatial deformation using a 2D spatial lookup process and 2D amplitude modulation. The 2D spatial lookup process involves translating four separate planes from a point of origin $(x_1, y_1), (x_3, y_3), (x_5, y_5), (x_7, y_7)$ translated by the movement of four separate streams $(x_2, y_2), (x_4, y_4), (x_6, y_6), (x_8, y_8)$.

The surface is also modulated by the relative direction and interactions of these four streams. A 2D terrain surface is generated iteratively based on the relative direction and distances between the four streams. Equation 3 describes this process for just two different streams (x_2, y_2) and (x_4, y_4). If the change in direction between these streams brings them closer together, an additive function is applied:

$$f(x,y)' = f(x,y) + \frac{\left(\left(\frac{x_2+x_4}{2}\right)x + \left(\frac{y_2+y_4}{2}\right)y + \sqrt{\left(x-\left(\frac{x_2+x_4}{2}\right)\right)^2 + \left(y-\left(\frac{y_2+y_4}{2}\right)\right)^2}\right)^{\frac{1}{100}}}{2\sqrt{\left(x-\left(\frac{x_2+x_4}{2}\right)\right)^2 + \left(y-\left(\frac{y_2+y_4}{2}\right)\right)^2}} \quad (3)$$

where $f(x,y)'$ is the new 2D function, and $f(x,y)$ is the previous 2D function. The iterative process is also applied subtractively for streams that are moving away from each other.

The terrain surface that is generated is then used to control the audio processing by using Wave Terrain Synthesis to control complex sound synthesis [16]. Similar techniques have been explored using Wave Terrain Synthesis as a framework for controlling timbre spatialisation in the frequency domain [17]. However, in this project, this approach it is used for controlling both granular synthesis and spectral spatialisation [18].

a. b.

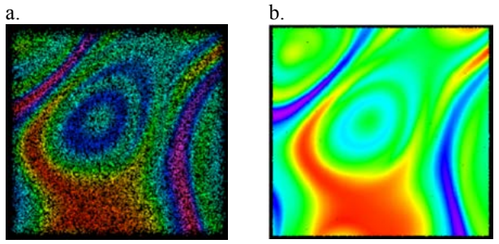


Figure 8 a. A trajectory of white noise reading values off the terrain after 1 second. b. A trajectory of white noise reading values off the terrain after 10 seconds.

The audio-rate trajectory that is used to read information from the terrain is a random 2D signal (white noise, as shown in Fig. 8), a curve that is considered to have effective space-filling properties. This means that details of the contour can be mapped to spatial details of the processing with great precision and resolution. The control information generated, in the way of 8192 individual parameters, those being 352,800 parameters generated per second, are used to control the relative distribution of grains and spectra across 8 loudspeakers.

Controlling granular synthesis via such an interface may take grain time or grain size into consideration. In order to control 1000 simultaneous grains, parameters would be updated at 44.1Hz. Depending on the implementation of the synthesis model, parameter assignments are multifarious. For example 2D data could determine the grain pan and grain length of individual grains.

Swarm-based spatialisation is also used where 2D data is mapped to the spatial position of individual grains. In this case the space-filling properties of the 2D trajectory signal will also correlate with the level of immersion of the resulting sound spatialisation.

Spectral spatialisation is also explored in Rodinia. Each spectral bin is assigned an independent spatial trajectory. 1024 simultaneous frequency bands are updated at lower-dimensional audio rates, that is, at approximately 43Hz. This is used to create complex immersive effects that would otherwise be more cumbersome if using standard control-rate methods.

5. CONTEXT

Preistly defines generative music as

> indeterminate music played through interaction between one or more persons and a more or less predetermined system, such that the players control some — but not all — performance parameters, and relinquish choices within a selected range to the system [19].

Tectonic: Rodinia conforms to this broadest definition of generative art work, through its use of algorithmically determined modification of the intentions of human conductors. The term most specifically refers here, however to the use of generative "emergent: non-repeatable" [20] music notation, a category of the emerging genre of animated notation [21].

It is an interactive form of generation that has game-like aspects to the conductors' interactions with the algorithmic modifications: a dynamic obstacle game. In this sense it resembles "4-way-confusion (4 agents)" games structure in which "four agents traveling in four opposing directions, meeting at nearly the same time [22] or (form the individual conductors perspective) a "Frogger"-like structure in which "one agent encounters many perpendicular crossing agents" [23].

The game analogy is perhaps amplified by the inclusion of an Audience View, allowing the audience both to hear and view the interactions of the streams, and the conductors' attempts to maintain control under conditions in which their choices are undermined and their ability to utilise the algorithmic modifications to subvert the control of the other conductors.

Musically, the work is something of a concerto for conductors themselves are silent but create sound through their gestures. The Rodinia environment gives significant freedom of choice to the conductors, which is curtailed only by the interactions between their choices.

6. CONCLUSIONS

Tectonic: Rodinia adds a series of new capabilities to the Decibel Scoreplayer. Many of these advances have been dependent upon the adoption of the Bonjour network protocol and the subsequent ability to stream data between a variety of devices.

There is arguably some value in engaging the audience with a visual representation of the sound they are hearing, but the requirements of the performer are quite different to those of the listener and displaying the performer's score to the audience and allowing them to "see what is coming" may reduce the effectiveness the musical discourse when it is actually heard. Delaying the audience score until the moment of its execution by the performers goes some way to alleviating the issue.

Rodinia is somewhat unusual in its combination of generative and interactive qualities in the context of notated music for live instrumentalists. Although the "tectonic" concept is distinct, the implementation of this work provides a framework capable of accommodating a wide range of generative and interactive/generative works employing varied conceptual approaches.

Acknowledgments

The XCode programming for *Tectonic: Rodinia* was developed by Aaron Wyatt. Many Thanks!

Partial funding for this project was provided by an Early Career Researcher Grant from Edith Cowan University.

7. REFERENCES

[1] C. Hope, A. Wyatt, and L. Vickery, "The Decibel ScorePlayer: New Developments and Improved Functionality", *Proceedings of the 2015 International Computer Music Conference*, Denton, 2015.

[2] L. Vickery, "Visualising the Sonic Environment", *Proceedings of the 2016* Electronic Visualisation and the Arts 2016, Canberra, 2016.

[3] L. Vickery, "An Approach to the Generation of Real-time Notation via Audio Analysis: The Semantics of Redaction", *Proceedings of the 2015 International Computer Music Conference*, Denton, 2015.

[4] K. Yarmosh, *App Savvy: Turning Ideas into iPad and iPhone Apps Customers Really Want*. O'Reilly Media, 2010, p. 53.

[5] L. Vickery, "Mobile Scores and Click-Tracks: Teaching Old Dogs", *Proceedings of the* 2010 *Australasian Computer Music Conference*, Canberra, 2010.

[6] E. Gilman & G. Underwood, "Restricting the Field of View to Investigate the Perceptual Spans of Pianists", *Visual Cognition* vol. 10, no. 2 pp. 201–32, 2003, p. 212.

[7] L. Vickery, "The Limitations of Representing Sound and Notation on Screen". *Organised Sound,* vol. 19, no. 3, 2014.

[8] L. Vickery, "Rhizomatic approaches to screen-based music notation". *forthcoming*, 2016.

[9] D. Chappell, "Taking a point for a walk: pattern formation with self-interacting curves", *Proceedings of Bridges 2014 Conference*, Tessellations, 2014, pp. 337–340.

[10] G. Greenfield, "Avoidance drawings evolved using virtual drawing robots", *Proceedings of EvoMUSART 2015*, Springer, 2015

[11] ibid p. 308

[12] L. Vickery, C. Hope, S. James, "Digital adaptions of the scores for Cage Variations I, II and III". *Proceedings of the 2012 International Computer Music Conference*, Ljubljana, pp. 426-432, 2012.

[13] D. Behrman, "What Indeterminate Notation Determines", *Perspectives on Notation and Performance*, pp. 74-89, Norton, 1976.

[14] P. Green-Armytage, "A Colour Alphabet and the Limits of Colour Coding", *Colour: Design*, 2010.

[15] D. Benedetti & E. Minto. "Tectonic Plate Simulation on Procedural Terrain", Retrieved from http://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S13/final_projects/benedetti_minto.pdf, 2013.

[16] S. James, "Spectromorphology and Spatiomorphology of Sound Shapes: audio-rate AEP and DBAP panning of spectra". *Proceedings of the 2015 International Computer Music Conference*, Texas, 2015.

[17] S. James, *Spectromorphology and Spatiomorphology: Wave Terrain Synthesis as a Framework for Controlling Timbre Spatialisation in the Frequency-Domain*, Ph.D Exegesis, Edith Cowan University, 2015.

[18] S. James, "A Multi-Point 2D Interface: Audio-rate Signals for Controlling Complex Multi-Parametric Sound Synthesis." *New Interfaces for Musical Expression*, 2016.

[19] J. Priestley. *Poiesthetic play in generative music*. PhD Virginia Commonwealth University, 2014.

[20] A. Biles, *GenJam in Transition: from Genetic Jammer to Generative Jammer*. *Proceedings* of *the 2002 International Conference on Generative Art*, Milan, 2002.

[21] P. Rebelo, Notating the unpredictable, Contemporary Music Review, vol. 29 no. 1, pp. 17-27, 2010

[22] S. Singh, M. Naik, M. Kapadia, P. Faloutsos, & G. Reinman, "Watch out! a framework for evaluating

steering behaviors". *Motion in Games*, pp. 200-209, Springer, 2008, p. 206.

[23] J. Henno, "On structure of games" in Information Modelling and Knowledge Bases XXI: Volume 206 Frontiers in Artificial Intelligence and Applications, 2010, p. 344.

AVA: A Graphical User Interface for Automatic Vibrato and Portamento Detection and Analysis

Luwei Yang[1] **Khalid Z. Rajab**[2] **Elaine Chew**[1]
[1] Centre for Digital Music, Queen Mary University of London
[2] Antennas & Electromagnetics Group, Queen Mary University of London
{l.yang, k.rajab, elaine.chew}@qmul.ac.uk

ABSTRACT

Musicians are able to create different expressive performances of the same piece of music by varying expressive features. It is challenging to mathematically model and represent musical expressivity in a general manner. Vibrato and portamento are two important expressive features in singing, as well as in string, woodwind, and brass instrumental playing. We present AVA, an off-line system for automatic vibrato and portamento analysis. The system detects vibratos and extracts their parameters from audio input using a Filter Diagonalization Method, then detects portamenti using a Hidden Markov Model and presents the parameters of the best fit Logistic Model for each portamento. A graphical user interface (GUI), implemented in MATLAB, allows the user to interact with the system, to visualise and hear the detected vibratos and portamenti and their analysis results, and to identify missing vibratos or portamenti and remove spurious detection results. The GUI provides an intuitive way to see vibratos and portamenti in music audio and their characteristics, and has potential for use as a pedagogical and expression analysis tool.

1. INTRODUCTION

Musicians introduce a high degree of acoustic variations in performance, above and beyond the categorical pitches and durations indicated in the musical score [1]. The sources of these acoustic variations include dynamic shaping, tempo variation, vibrato, portamento, staccato, and legato playing. While some expressions have been notated in the score (e.g. tempo and dynamics), musicians sometimes alter the instructions to create their own expressions [2]. We call these devices expressive features as they are usually not denoted in the composition but adopted in performance. These devices result in unique performance styles that differentiate one musician from another.

We focus on two expressive features: vibrato and portamento. Vibrato is a periodic modulation of frequency, amplitude, and even spectrum [3]. Portamento is the note transition that allows musicians to adjust the pitch continuously from one note to the next [4]. Vibrato and portamento characteristics can be used to reveal differences in performance styles, and performance variation among different musicians [4, 5, 6, 7, 8].

This paper presents an off-line system, AVA, which accepts raw audio and automatically tracks the vibrato and portamento to display their expressive parameters for inspection and further statistical analysis. We employ the Filter Diagonalization Method (FDM) to detect vibrato [9]. The FDM decomposes the local fundamental frequency into sinusoids and returns their frequencies and amplitudes, which the system uses to determine vibrato presence and vibrato parameter values. A fully connected three-state Hidden Markov Model (HMM) is applied to identify portamento. The resulting portamenti are modeled as Logistic Functions which are well suited to displaying the characteristics of a portamento [4]. The AVA system has been implemented in MATLAB and consists of a graphical user interface (GUI) and all relevant functions [1].

The structure of the paper is as follows: Section 2 presents the vibrato and portamento feature detection and analysis modules. Section 3 introduces AVA's MATLAB interface, and Section 4 presents discussions and conclusions.

2. FEATURE DETECTION AND ANALYSIS

The basic architecture of the AVA system is shown in Figure 1. Taking the audio as input, the pitch curve (fundamental frequency) is extracted using the pYIN method [10], a probabilisitic version of the original Yin method[11]. The resulting pitch curve is sent to the vibrato detection module, which identifies vibrato existence using an FDM-based method. The detected vibratos are forwarded to the module for vibrato analysis, which outputs the vibrato statistics. To ensure the best possible portamento detection performance, we flatten the detected vibratos using the built-in MATLAB 'smooth' function as the oscillating shape of the vibrato degrades portamento detection. The HMM-based portamento detection module uses this vibrato-free pitch curve to identify potential portamenti. A Logistic Model is fitted to the detected portamentos for quantitative analysis. Moreover, if there are errors in detection, the interface allows the user to indicate missing vibratos or portamenti and remove spurious results.

2.1 Vibrato Detection and Analysis

There exist two kinds of vibrato detection methods: note-wise and frame-wise methods. Note-wise methods require

[1] The beta version of AVA is available at luweiyang.com/research/ava-project.

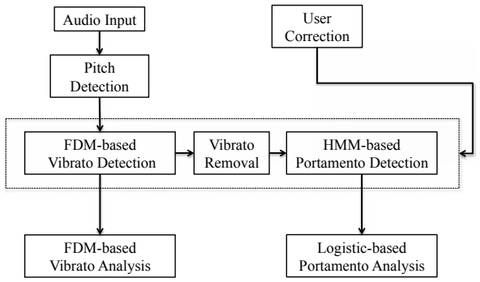


Figure 1. The AVA architecture.

Algorithm 1: The FDM algorithm

Input: Pitch curve (fundamental frequency)
Output: The frequency and amplitude of the sinusoid with the largest amplitude
Set the scan frequency range;
Filter out any sinusoids whose frequency is out of the frequency range of inteterst;
Diagonalize the matrix formed by the pitch curve;
for *each itertation* **do**
  Create a matrix using 2D FFT on the pitch curve;
  Diagonalize this matrix;
  Get eigenvalues;
  Check the acceptance of eigenvalues;
end
Calculate the frequencies from the eigenvalues;
Calculate the amplitudes from the corresponding eigenvectors;
Return the frequency and amplitude of the sinusoid with the largest amplitude;

a note segmentation pre-processing step before determining if the note contains a vibrato [12, 13]. Frame-wise methods divide the audio stream, or the extracted pitch curve information, into a number of uniform frames. Vibrato existence is then decided based on information in each frame [14, 15, 16, 9].

We employ the Filter Diagonalization Method (FDM) described in [9] to detect vibratos and characterize their properties. The FDM is able to extract the frequency and amplitude of sinusoids for a short time signal, thus making it possible to determine vibrato presence over a short time span. Fundamentally, the FDM assumes that the time signal (pitch curve) of a frame is the sum of exponentially decaying sinusoids,

$$f(t) = \sum_{k=1}^{K} d_k e^{-in\tau\omega_k}, \text{ for } n = 0, 1, \ldots, N, \qquad (1)$$

where K is the number of sinusoids required to represent the signal to within some tolerance threshold. ω_k and d_k are fitting parameters which are defined as the complex frequency and complex weight, respectively, of the k-th sinusoid. The aim of the FDM is to find the $2K$ unknowns, representing all ω_k and d_k. A brief summary of the steps is described in Algorithm 1. Details of the algorithm and implementation are given in [9]. Here, we only consider the frequency and amplitude of the sinusoid having the largest amplitude. A Decision Tree is applied to these two parameters to determine vibrato presence. The window size is set to 0.125 seconds and step size is one quarter of the window. Note pruning (throw away any feature whose duration is less than a threshold) used a threshold of 0.25 seconds.

The vibrato rate and extent fall naturally out of the FDM analysis results. In addition, to characterize the shape of a detected vibrato, we use sinusoid similarity as described in [7]. The sinusoid similarity is a parameter between 0 and 1 that describes the similarity of a vibrato shape to a reference sinusoid using cross correlation.

2.2 Portamento Detection and Analysis

To detect portamentos, we have created a fully connected three-state HMM using the delta pitch curve as input as shown in Figure 2. The three states are down, steady, and up, which correspond to slide down, steady pitch, and slide up gestures. Based on experience, we estimate the transition probabilities to be those shown in Table 1. A Gamma distribution models the probability density distribution of each down and up state's observation. The steady pitch observation's probability density distribution is modeled as a sharp needle around 0 using a Gaussian function. The best likely path is decoded using the Viterbi algorithm. All state changes are considered as boundaries. The 0.09 second note pruning is applied.

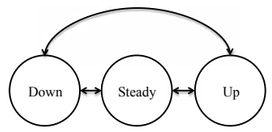


Figure 2. The portamento detection HMM transition network.

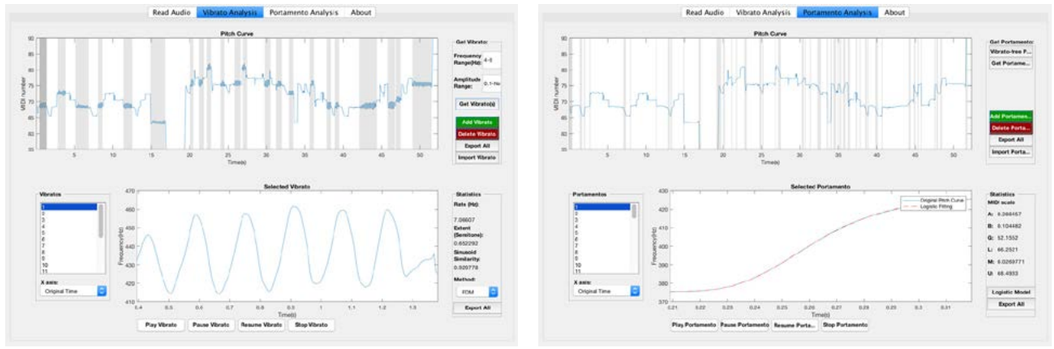
	Down	Steady	Up
Down	0.4	0.4	0.2
Steady	1/3	1/3	1/3
Up	0.2	0.4	0.4

Table 1. Transition probability of HMM-based portamento detection.

To quantitatively describe the portamento, we apply the Logistic Model in the fashion described in [4]. This model is motivated by the observation that portamenti largely assume S-shapes. An ascending S shape is characterized by an acceleration in the first half and a deceleration in the second half. An inflection point exists between these two processes. The Logistic Model is described as

$$P(t) = L + \frac{(U - L)}{(1 + Ae^{-G(t-M)})^{1/B}}, \qquad (2)$$

where L and U are the lower and upper horizontal asymptotes, respectively. Musically speaking, L and U are the



(a) Vibrato Analysis (b) Portamento Analysis

Figure 3. Screenshots of AVA.

antecedent and subsequent pitches of the transition. A, B, G, and M are constants. G can further be interpreted as the growth rate, indicating the steepness of the transition slope. The time of the inflection point is given by

$$t_R = -\frac{1}{G} \ln\left(\frac{B}{A}\right) + M \,. \qquad (3)$$

The pitch of the inflection point can then be calculated by substituting t_R into Eq. (2).

3. THE AVA INTERFACE

The vibrato and portamento detection and analysis methods described above have been implemented in MATLAB. AVA's GUI consists of three panels accessed through tabs: Read Audio, Vibrato Analysis, and Portamento Analysis. The Read Audio panel allows a user to input or record an audio excerpt and obtain the corresponding pitch curve. The Vibrato Analysis and Portamento Analysis panels provide visualizations of vibrato and portamento detection and analysis results, respectively.

The left screenshot in Figure 3 shows the Vibrato Analysis panel analyzing an erhu excerpt. The pitch curve of the entire excerpt is presented in the upper part, with the shaded areas indicating possible vibratos. Vibrato existence is determined using the FDM-based vibrato detection method, which is triggered using the button in the upper right. The interface allows the user to change the default settings for the vibrato frequency and amplitude ranges; these adaptable limits serve as parameters for the Decision Tree vibrato existence detection process.

Shaded boxes highlight the detected vibratos on the pitch curve. Two edit functions, allowing the adding and deleting of shaded windows indicating detected vibratos, are provided for users to correct vibrato detection errors. On the lower left is a box listing the indices of the detected vibratos. The user can click on each shaded area, or choose element in the listing box, or use the left- or right-arrow keys, to navigate between vibratos. The selected vibrato pitch curve is presented in the lower plot with corresponding parameters shown on the right. In this case, with the vibrato frequency range threshold $[4, 9]$ Hz and amplitude range threshold $[0.1, \infty]$ semitones, the selected vibrato has frequency 7.07 Hz, extent 0.65 semitones, and sinusoid similarity value 0.93. A drop down menu allows the user to choose between the original time axis and a normalized time axis for visualizing each detected vibrato. A playback function assists the user in vibrato selection and inspection.

The right plot in Figure 3, shows the Portamento Analysis panel for the same music excerpt. The pitch curve shown here is that of the Vibrato Analysis panel after flattening the detected vibratos so as to improve portamento detection. Like the Vibrato Analysis panel, the Portamento Analysis panel also provides add and delete functions for the shaded windows indicating the detected portamenti. A click of a button initiates the process to fit Logistic Models to all the portamenti.

The best-fit Logistic model is shown as a red dashed line against the original portamento pitch curve. A panel to the right shows the corresponding Logistic parameters. In the highlighted case, the growth rate is 52.16 and the lower and upper asymptotes are 66.25 and 68.50 (in MIDI number), respectively, which could be interpreted as the antecedent and subsequent pitches. From this, we infer that the transition interval is 2.25 semitones.

Our design principle was to make each panel provide a core functionality while minimizing unnecessary functions having little added value. As vibratos and portamenti relate directly to the pitch curve, each tab shows the entire pitch curve of the excerpt and a selected vibrato or portamento in that pitch curve. To enable user input, we allow the user to create or delete feature highlight windows against the pitch curve. Playback functions allow the user to hear each detected feature so as to inspect and improve detection results. To enable off-line statistical analysis, AVA can export to a text file the vibrato and portamento annotations and the corresponding parameters.

4. DISCUSSIONS AND CONCLUSIONS

In this paper, we have presented an off-line automatic vibrato and portamento detection and analysis system. The system implements an FDM-based vibrato detection method

and an HMM-based portamento detection method. Vibrato parameters is a natural by-product of the FDM process, and a Logistic Model is fitted to each portamento. The system has been implemented in MATLAB, and the GUI provides intuitive visualization of detected vibratos and portamenti and their properties. User feedback allows for the correction of false positive and false negative errors.

The vibrato detection module currently uses a Decision Tree method for determining vibrato existence. The user can set the vibrato frequency and amplitude ranges to affect the output. A more sophisticated Bayesian approach taking advantage of learned vibrato rate and extent distributions is described in [9]. The distributions can be adapted to each instrument or music genre. While this method has been shown to give better results, it requires training data beforehand.

The portamento detection method sometimes misclassifies normal note transitions as portamenti even though a minimum duration threshold is used to prune the results. We observe that the false positives tend to have low intensity (dynamics) values. Future improvements to the HMM-based portamento detection method could take into account intensity features in addition to the delta pitch curve.

The applications of AVA include music education and expression analysis, and its outputs provide a useful base for expression synthesis and transformation. In instrument learning, AVA can be used to provide visual feedback and quantitative analysis of students' performances, allowing students to inspect their expressive features and adapt accordingly. Furthermore, it is able to integrate other modules, e.g. intonation detection, rhythm detection, to be a completed pedagogical tool. In expression analysis, AVA can be used to quantify musicians' vibrato/portamento playing styles, and the ways in which they use these expressive features. When applied to music of different cultures, it is able to be used to conduct large-scale comparative studies. The analysis results from AVA can also serve as input to expression synthesis engines, or to transform the expressive features in recorded music.

Acknowledgments

This project is supported in part by the China Scholarship Council. The authors would like to thank Siying Wang for discussions of the MATLAB interface.

5. REFERENCES

[1] C. Palmer and S. Hutchins, *What is musical prosody?* ELSEVIER, 2006, ch. 46, p. 245.

[2] K. Kosta, O. F. Bandtlow, and E. Chew, "A Change-Point Approach Towards Representing Musical Dynamics," in *Mathematics and Computation in Music*. Springer, 2015, pp. 179–184.

[3] V. Verfaille, C. Guastavino, and P. Depalle, "Perceptional Evaluation of Vibrato Models," in *Proceedings of the Conference on Interdisciplinary Musicology(CIM05)*, March 2005.

[4] L. Yang, E. Chew, and K. Z. Rajab, "Logistic Modeling of Note Transitions," in *Mathematics and Computation in Music*. Springer, 2015, pp. 161–172.

[5] T. L. Nwe and H. Li, "Exploring Vibrato-Motivated Acoustic Features for Singer Identification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 2, pp. 519–530, 2007.

[6] T. H. Özaslan, X. Serra, and J. L. Arcos, "Characterization of embellishments in ney performances of makam music in turkey," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2012.

[7] L. Yang, E. Chew, and K. Z. Rajab, "Vibrato Performance Style: A Case Study Comparing Erhu and Violin," in *Proc. of the 10th International Conference on Computer Music Multidisciplinary Research (CMMR)*, 2013.

[8] H. Lee, "Violin portamento: An analysis of its use by master violinists in selected nineteenth-century concerti," in *9th International Conference on Music Perception and Cognition, ICMPC9 Proceedings of*, August 2006.

[9] L. Yang, K. Z. Rajab, and E. Chew, "Filter Diagonalisation Method for Music Signal Analysis: Frame-wise Vibrato Detection and Estimation," *Journal of Mathematics and Music*, 2016, under revision.

[10] M. Mauch and S. Dixon, "pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, 2014.

[11] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, pp. 1917–1930, 2002.

[12] S. Rossignol, X. Rodet, P. Depalle, J. Soumagne, and J.-L. Collette, "Vibrato: detection, estimation, extraction, modification," in *Proceedings of the 2nd COST-G6 Workshop on Digital Audio Effects (DAFx)*, December 1999.

[13] H.-S. Pang and D.-H. Yoon, "Automatic detection of vibrato in monophonic music, Pattern Recognition," *Pattern recognition*, vol. 38, p. Pattern recognition, 2005.

[14] P. Herrera and J. Bonada, "Vibrato extraction and parameterization in the spectral modeling synthesis framework," in *Proceedings of the Digital Audio Effects Workshop (DAFX98)*, 1998.

[15] J. Ventura, R. Sousa, and A. Ferreira, "Accurate analysis and visual feedback of vibrato in singing," in *Proceedings of the 5th International Symposium on Communications, Control and Signal Processing*, May 2012, pp. 1–6.

[16] H. von Coler and A. Roebel, "Vibrato Detection Using Cross Correlation Between Temporal Energy and Fundamental Frequency," in *Audio Engineering Society Convention 131*, October 2011.

Spectrorhythmic evolutions: towards semantically enhanced algorave systems

Alo Allik
Queen Mary University of London
a.allik@qmul.ac.uk

ABSTRACT

This paper explores enhanced live coding as a strategy for improvisatory audiovisual performances of rhythm-based music. The real time decision-making process of the programmer-performer is informed and aided by interactive machine learning, artificial intelligence and automated agent algorithms. These algorithms are embedded in a network-based distributed software architecture of an audiovisual performance system, which is comprised of computer graphics, sound synthesis and algorithmic composition clients. The system facilitates human-computer interaction through live coding during performances to create extemporized immersive multimodal experiences for audiences. The audiovisual content during performances is created with reactive artificial life algorithms, evolutionary sound synthesis, machine listening and music analysis. Autonomous agent systems, audio feature extraction and linked semantic data formats help the performer cope with the complexities of live coding multimedia performance environments.

1. INTRODUCTION

We live in a cultural environment in which computer based musical performances enhanced by multimedia have become ubiquitous. Particularly the use of laptops as instruments is a thriving practice in many genres and subcultures. The opportunity to command the most intricate level of control on the smallest of time scales in music composition and computer graphics introduces a number of complexities and dilemmas for the performer working with algorithms. Writing computer code to create audiovisuals offers abundant opportunities for discovering new ways of expression in live performance while simultaneously introducing challenges and presenting the user with difficult choices. There are a host of computational strategies that can be employed in live situations to assist the performer, including artificially intelligent performance agents who operate according to predefined algorithmic rules. This paper describes a software system for real time audiovisual improvisation and composition in which live coding as a computational strategy for audiovisual laptop performances is explored. The features of the framework intend to enhance the live coding practice by analysis of traditional music, evolutionary computing, reactive computer graphics, and linked data technologies. The name of the performance environment - *sparsematrix* - is derived from a method of musical analysis applied to traditional African music which calculates sparse representations of rhythm patterns in order to capture their skeletal time structures. For sound synthesis, an evolutionary algorithm is utilised that enables evolving large populations of complex synthesis graphs, either in real time or for later reuse. The structure of the evolutionary synthesis process is described in a light-weight OWL ontology, while the graphs are stored in a CouchDB database [1] and linked to the ontology using JSON-LD [2], a semantic extension of the standard JSON format. The computer graphics component implements a 3-dimensional world of cellular automata that operates in parallel as a self-organising map of audio analysis vectors in Cinder [3], an open source OpenGL library in C++.

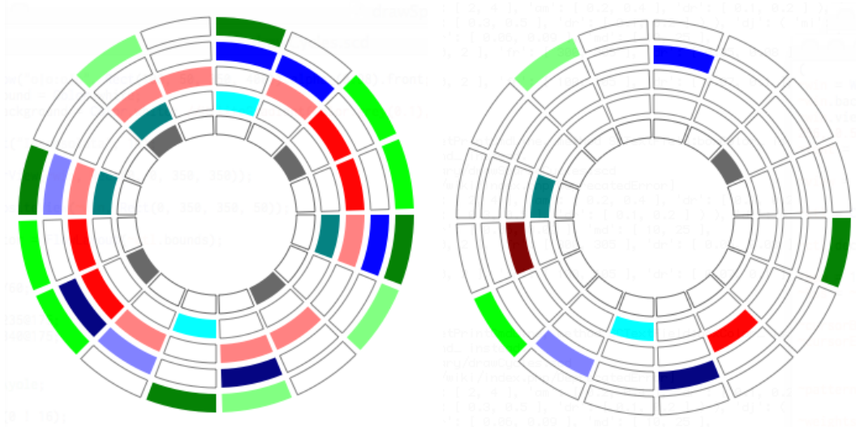
Live coding has been firmly established as an improvisatory musical performance practice. Musicians-programmers have been exploring alternative methods to user-interface-driven production software in order to explore more spontaneous and flexible ways to compose music in real time (see for example [1] for a review of different live coding practices that existed already over a decade ago). At the same time, the advances in computing speed and power are constantly shifting the boundaries of what is possible to achieve with sound synthesis, algorithmic composition and performance interfaces in the context of real time interactive music systems. Arguably this is the main factor behind the proliferation of real time information systems used in musical performances. However, the abundance and complexity of musical algorithms constitute a specification problem for the artist, who is constantly faced with difficult choices regarding levels of granularity for musical parameter control. Writing a low level sound generating function live on stage does not necessarily make for a captivating or even intellectually invigorating experience for the audience or, arguably, the performer. At the other end of this imaginary spectrum, graphical user interfaces of proprietary music production applications enforce preconceived and often limited compositional principles upon users, seriously stifling the latent potential of algorithmic creativity. The optimal parameter control level lies somewhere in between these two extremes.

2. STRUCTURE OF THE ENVIRONMENT

The basic architecture of the *sparsematrix* environment consists of

- algorithmic composition libraries developed in the SuperCollider[2] programming environment

[1] http://couchdb.apache.org
[2] http://json-ld.org
[3] https://libcinder.org



(a) Original Yole pattern (b) Yole pattern after application of sparsity rules

Figure 1. Cyclic representation of sparsity rules

- a precompiled graphics application developed in C++ using the Cinder library
- a CouchDB JSON-LD database
- a Semantic Web ontology that describes the synthesis environment

The modular client-server structure of SuperCollider is primarily intended for real time applications. The communication between server and client happens through the Open Sound Control (OSC) [4] protocol for communicating musical information over a TCP or UDP network. SuperCollider is a dynamic programming language which enables using live coding as an interactive programming paradigm to explore synthesis and composition algorithms in real time. The *sparsematrix* environment implements a library of composition algorithms including a dictionary of traditional djembe rhythm patterns, functions that apply various rules to convert the patterns to sparse arrays, classes that facilitate efficient retrieval of gene expression synthesizers from the CouchDB database, mapping of the synthesizers to the rhtyhm patterns and controlling their various parameters (amplitudes, durations, effects sends, etc.) through live coding, sound spatialization tools using the Ambisonic Toolkit [5] , and real time machine listening algorithms. The musical analysis method used to derive rhythm patterns for structuring the musical output of the system is described in Section 3. The machine listening algorithms combine real time onset detection with timbral feature extraction, so that when the system detects a musical event in the real time audio output, a feature vector is sent to the graphics application via OSC and used to trigger visual content. The environment is designed as a modular architecture of specialized components to facilitate extensibility and distributed deployment. The following 4 sections describe these components in more detail, beginning with algorithmic composition and sound synthesis modules, then the JSON data storage solution, followed by the computer graphics application.

[4] http://opensoundcontrol.org/introduction-osc
[5] http://www.ambisonictoolkit.net/

3. RHYTHM ANALYSIS

The central compositional concept in the system is the sparse representation of traditional rhythms in order to capture their basic time structure. This approach was partly inspired by Carsten Nicolai's - also known by the artist name Alva Noto - sonic experiments in which he "tried to strip bare pop music standards to a minimum level of rhythmic structure"[3], that produced remarkable results in terms of musical form. The concepts developed in this system rely on computational analysis and simple combinatorics. The algorithm implemented is not optimal and often does not find a solution that satisfies the constraints, however, it does preserve the most essential structure of the patterns, which can be used in the performance as a structuring element. African rhythm patterns were chosen as the raw material for their rhythmic complexity, as well as the rich and ancient cultural tradition, which have been historically often ignored in musicological research.

Fig. 1 shows the rhythm patterns in cyclic notation before (on the left) and after (on the right) the sparse algorithm is applied. The alternative representation has been developed based on geometric representations of cyclic rhythms previously used in ethnomusicological research[4][5]. The example is based on a rhythm pattern called Yole from a library of African rhythms which have been transcribed from the original notation for 5 instruments: 3 djembe drums, 1 bass drum and a bell. Each of the djembe drums is represented in the notation as producing 3 distinct sounds: bass, tone and slap, each of which in turn is considered a different instrument in the algorithm. This is how we end up with 11 parallel patterns: the Yole pattern specifies 3 distinct sounds for the first 2 djembes and just 2 distinct sounds for the third djembe and the bass, while the bell sound does not vary and is represented as the 11th instrument. The sparse representation is achieved by applying simple constraints to the matrix of events, not allowing an

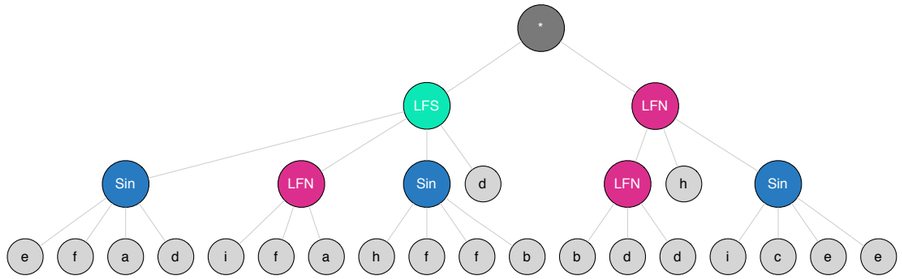


Figure 2. A simple synthesis graph with nesting depth 2 where "LFS" represents a sawtooth oscillator, "Sin" a sinewave oscillator, "LFN" a low-frequency noise generator, and lower case letters are used to represent floating point parameters to synthesis functions.

event to share the same row or column with another event. In order to preserve the most salient features of the pattern, the importance of each time unit is determined by the number of concurrent events in the original pattern. The algorithm then needs to find the best compromise between preserving the most important time units and keeping all instruments represented in the sparse matrix. The sparsification algorithm is designed to calculate alternative solutions as sub-patterns to increase rhythmic complexity. These sub-patterns can be appended to the main patterns in either dimension.

4. GENE EXPRESSION SYNTHESIS

The synthesizers mapped to event streams are created with an evolutionary algorithm. Gene expression synthesis (GES)[6] is a way to evolve sound synthesizers in computer code. These synthesizers are computer programs that produce sound when executed. Gene expression synthesis uses the methods of gene expression programming proposed by Candida Ferreira[7]. Gene expression programming combines ideas from the two most widely used and well-known evolutionary programming methods: genetic algorithms and genetic programming. Genetic algorithms encode solutions to programming problems as linear strings of digits, typically binary, while genetic programming evolves computer programs as tree structures in which every computer function or parameter value is represented as a node. Gene expression programming does both: the solutions are encoded as linear strings which are decoded into tree structures representing computer programs. Each candidate solution defined as a chromosome consists of codons - elemental units of GES - that can be either functions or terminals. The translation from genotype to phenotype follows a simple, breadth-first recursive principle: as the codons of a gene are traversed, for each function encountered, the algorithm reserves a number of following unreserved codons as arguments to that function regardless whether they are functions or terminals. Once decoded as an executable program, each candidate solution in a population is subjected to a fitness evaluation that determines how well it performs at solving a target problem.

The computer programs evolved with GES are sound synthesis graphs in this case implemented in the SuperCollider programming environment. Each solution generates a SuperCollider SynthDef object. The functions in a GES chromosome are Unit Generators, sound generating functions that serve as basic building blocks of synthesis graphs. Fig. 2 shows a simple example of a synthesis graph. Fitness is primarily evaluated in terms of a distance metric of features from target audio, while secondary methods are used to ensure the structural and functional integrity of each synthesis graph as well as balancing factors between resource efficiency and graph complexity. There are various phases of fitness evaluation due to added complexity inherent in digital audio. Programmatic validity phase ensures that only the candidate solutions that successfully pass compilation are considered for the subsequent phase. Functional validity check excludes synthesizers that produce invalid audio output (i.e. nil, inf, NaN). The main fitness function measures the distance of audio feature vectors between the candidate synthesizer and a target sound, which is selected by the user depending on the context of the experiment. The features currently used include mean and standard deviation values for 20 Mel-Frequency Cepstral Coefficients (MFCC), Spectral Flatness, Spectral Centroid, and Amplitude. See [8] for a thorough overview of commonly used audio features. Resource efficiency measure has been implemented in order to imitate the condition of limited resources of natural selection, so each candidate solution is assigned a CPU usage value measured during the execution of the synthesizer. This selection pressure introduces a tendency in the population of favouring simpler synthesizer graphs over more complex ones. To counteract this tendency, a conflicting fitness pressure is introduced to encourage structural complexity in the form of rewarding greater nesting depth. This way, the complexity can be maintained in populations, while still encouraging resource usage effectiveness.

Once each individual has been assigned a fitness value, the population is subjected to various standard genetic operators, including **replication** (weighted random selection of individuals for next generation according to fitness), **mutation** (a random change of a single codon value based on a user defined parameter), **transposition** (copying of codon sequences to different locations on a chromosomes), and **recombination** (exchange of genetic material between chromosomes).

5. SYNTHESIZER DATABASE

Due to large volumes of data potentially involving thousands of candidate solutions deemed suitable for performances, GES synthesizers with their associated metadata are stored in a CouchDB database as JSON data structures. This includes all the data necessary to reconstruct each synthesizer for use during a performance or as sources for subsequent evolutionary synthesis experiments. The synthesiser metadata is structured using a light-weight OWL ontology, shown in Fig. 3, that enables representation of GES synthesizers on the Semantic Web and linking of the synthesizer audio features to the concepts defined in the Audio Feature Ontology framework [6]. This framework is being developed as part of a larger effort to represent music-related concepts using Semantic Web technologies and formats. The Audio Feature Ontology framework is part of a library of modular ontologies and enables representation of common concepts in the domain of music information retrieval. The synthesizer data that is communicated between CouchDB and SuperCollider is expressed in terms of the GES ontology using JSON-LD[9]. Listing 1 shows a fragment of a GES synthesizer data structure in JSON format as stored in CouchDB. The embedded context defines the relevant namespaces so that the GES ontology classes and properties can be meaningfully expressed in JSON and concepts from external sources can be linked. In this example, SpectralCentroid and SpectralFlatness audio features are defined in the Audio Feature Vocabulary, so it becomes possible to query more information about these features from the external source, for example, finding information about how these particular features are computed by querying steps of computational workflows. However, this is just the first step in attributing structure and meaning to the featureless synthesiser space. One of the options might be developing a semantic concept framework based on crowd-sourced tags from a collaborative open sound database like Freesound [7] using feature comparison.

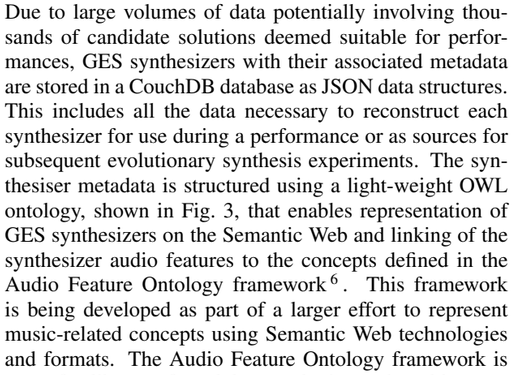


Figure 3. Structure of the gene expression synthesis ontology

The gene expression synthesis algorithm can be used for isolated experiments to generate populations of synthesizers for reuse during performances. However, the *sparse-*

[6] http://sovarr.c4dm.eecs.qmul.ac.uk
[7] http://www.freesound.org

```
{
   "@context": {
      "ges": "http://geen.tehis.net/ontology/"
   },
   "_id": "1a75f4f87bdcac24b6ba5fc25c003ab2",
   "@type": "ges:Synth",
   "ges:environment": {
      "@type": "ges:Environment",
      "ges:headsize": 24,
      "ges:numgenes": 1,
      "ges:linker": {
         "@type": "ges:Function",
         "ges:name": "*",
         "ges:class": "AbstractFunction"
      }
   },
   "ges:defname": "gep_gen000_061_141212_225728",
   "ges:genome": [ "LFPar", "LFGauss", "SinOsc",
      "v", "PMOsc", "e", "j", "a", "c", "a"],
   "ges:features": {
      "ges:centroid": {
         "ges:mean": 526.07188020057,
         "ges:std_dev": 161.03829149232
      },
      "ges:flatness": {
         "ges:mean": 0.032055785092016,
         "ges:std_dev": 0.017184103858522
      }
   }
}
```

Listing 1. Fragment of a GES synthesizer JSON structure

matrix environment also enables the performer to evolve and play synthesizers live on stage. The semantically enriched live coding environment aids the performer in the decision making process when selecting synthesizers from the database or evolving them in real time. The synthesizers are classified according to audio feature vectors and different maps are created to visualize the distribution of feature vectors from different perspectives. For example, a plot, shown in Fig. 4, of spectral flatness - how noise-like a signal is - against spectral centroid - how bright the sound is - gives an idea about the characteristic of each synthesizer. This can be further aided by classifying synthesizers with a self-organised 2-dimensional map of MFCC vectors. The performer can make informed selections of synthesizers either for use in the real time live coding composition process or as source material for real time evolution by going through the iterations of the GES algorithm.

6. REACTIVE COMPUTER GRAPHICS

The computer graphics are based on two autonomous computational systems - cellular automata and self-organizing maps - which are reactive to the audio analysis data communicated to the graphics application. The cellular automata system implements rules known as outer totalistic cellular automata[10]. Outer totalistic rules belong to the category of discrete-state automata and are a subset of what are known as totalistic rules. Totalistic rules are defined in terms of only the total or average of the values of the cells in a neighborhood. Outer totalistic rules consider the center cell state separately from the outer total, i.e. the total of the neighborhood. Another feature built into the rule definitions is the option to add decay when defining a rule,

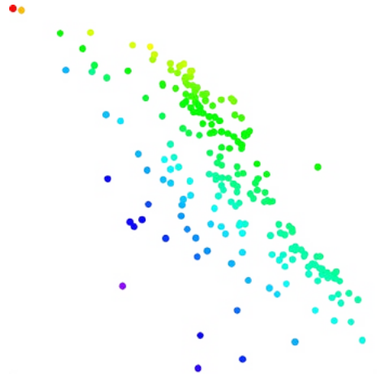


Figure 4. An example of a feature map with Spectral Flatness on the x-axis plotted against Spectral Centroid

inspired by the *Generations* rule family from the MCell automata lexicon[11]. Decay means when a cell "dies", it does not immediately switch to state 0, but goes through a number of discrete intermediate steps before reaching state 0. The rules are initialized by defining 3 parameters:

- An array of totals of neighborhood cells in state 1 that cause the center cell to transition from 0 to 1 or "birth" rules. These are only applicable if the center cell is in state 0.
- An array of totals of neighborhood cells in state 1 that maintain the center cell's state 1 or "survival" rules. These are only applicable if center cell is in state 1. If the array does not contain the total, then the center cell's state transitions from 1 to 0.
- the number of intermediate steps when the cell transitions from 1 to 0 or "decays".

In order to generate patterns of behavior, an initial configuration of cells has to be specified and this can be done in many different ways. In the *sparsematrix* environment, each cell acts as a node in a Self-Organizing Map (SOM) of live audio analysis data. A self-organizing map (SOM) is a type of artificial neural network proposed by Teuvo Kohonen[12] that is trained with sample data in form of input vectors using unsupervised learning. Once the map is trained, it can be used to classify input vectors. Each node represents a vector of weight values. The map is initiated by vectors containing random values. For each iteration of new input vector, the system proceeds through 2 phases:

- Find the similarity of the input vector for each node in the map using Euclidean distance, retaining the smallest distance value which is called the Best Matching Unit (BMU)
- For all nodes within a defined radius of the BMU, update the node's weight vector scaled by the distance from the BMU.

Once the iteration is complete, the SOM can be used for classification of input vectors, which in the *sparsematrix* system originate from live signal processing analysis. The map defines a location of each input vector using the BMU identification step in the three-dimensional space. The SOM is trained in real time during the performance with MFCC data vectors received from the audio analysis of the live input. Each time the Best Matching Unit (BMU) is determined, the corresponding cell in the automata world is activated together with its diametrically opposite counterpart in each dimension. Figure 5 shows a screenshot of

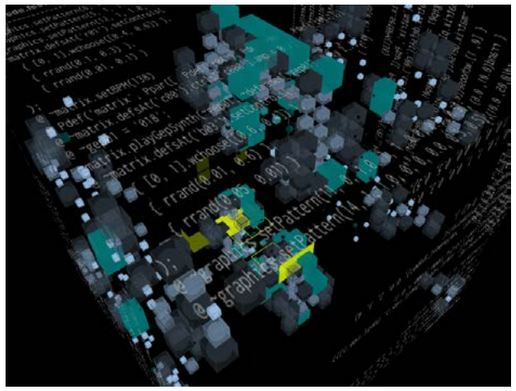


Figure 5. *sparsematrix* graphics with the evaluated code mapped onto the edges of the world.

the graphics application computing the above defined complex rule. There are 3 OpenGL graphics patterns activated mapped to visualize system dynamics with different parameters. The floating text visible in the figure is the code that has been evaluated during the session and mapped to the edges of the cube-shaped world in an attempt to incorporate it into the visual aspect of the performance while finding alternatives to the widely accepted live coding practice of projecting the programming environment to the screen.

7. LIVE CODING

The sparsified rhythm patterns described in Section 3 are used in live performances to structure the musical material by mapping sound synthesis functions generated with GES to events in the patterns. The system provides a live coding interface in the shape of object-oriented programming structures that expose different levels of parameter control to the user. Each pattern consists of a number of parallel event streams (typically between 32 and 64) which can be activated individually. Each event stream has a number of parameters associated with it. In addition to the already mentioned synthesizer mapping and activation, the user can control amplitude, duration, frequency, rotation in 3-dimensional Ambisonic field (rotate, tilt, and tumble), shape of the amplitude envelope, audio effect selection and the amount of the signal sent to the audio effect processor. This offers a large number of controllable parameters to the user. Considering that there are 20 patterns in the database, each of which could contain over 30 event streams depending on the sparsification parameters selected, the total number of controllable streams easily reaches hundreds at any point in a performance. Parameter control can be re-

alized by accessing each of the event stream control variables individually, which may prove inefficient when dealing with a large amount of options. Alternatively a more efficient strategy is to use custom written or built-in iterative probability functions inside time-based routines to access groups of parameters simultaneously.

Another real time compositional method available to the programmer-performer involves the statistical methods used to classify the synthesisers by spectral descriptors when deciding the mapping of synthesis functions to events as described in Section 5. This enables exploring global operations like homogenization and diversification of the spectral content of the performance, that is, choosing synthesisers from a particular small region or cluster on a feature map as opposed to from the largest region possible. This approach enables improvisation with spectral dynamics or, in other words, exploring timbre space as a musical control structure.

8. DISCUSSION AND CONCLUSIONS

The general problem that laptop artists encounter is how to specify levels of granularity for musical and visual parameter control. The general problem that the development of the *sparsematrix* environment is striving to solve is how to define parameter control on multiple time scales and levels of compositional hierarchy and what are the most efficient and flexible strategies for controlling these parameters during a performance. The motivations behind the project include the current state and the potential of open source software and programming tools offer a host of enticing possibilities for artists working with interactive digital multimedia. The proliferation of interactive digital art developed with these free and open technologies might be implying shifting social attitudes towards the role of technology and their impact on society and culture. The open source aesthetic in itself has come to signify a very different social attitude in the context of the intellectual property rules of the dominant corporate socioeconomic model. In the electronic music domain it may also be considered as signaling an adjustment from the academic tradition of electroacoustic tape music towards real time and interactive models of musical composition. Live coding as an interface for real time composition enables the artist to explore compositional features of the system without being confined to a graphical interface that typically limits the available options and enforces a predetermined compositional aesthetic. The choices of parameter control granularity in the *sparsematrix* live coding system provide means of exploring rhythmic and spectral complexity in the context of Algoraves [8], where the focus is on humans making and dancing to music. The environment has a flexible and modular cross-platform architecture. This has enabled to adopt the system for different setups depending on availability of computing resources. The system has been successfully deployed at various public events and festivals internationally including in the UK, Germany, Spain, Belgium, Greece, Switzerland and Japan. The most recent performances have utilized 2 laptop computers in order to separate graphics processing from the rest of the system for increased computational resources.

[8] http://algorave.com

9. REFERENCES

[1] N. Collins, A. McLean, J. Rohrhuber, and A. Ward, "Live Coding in Laptop Performance," *Organised Sound*, vol. 8, no. 3, pp. 321–330, 2003.

[2] J. McCartney, "Rethinking the Computer Music Language: SuperCollider," *Computer Music Journal*, vol. 26, no. 4, 2002.

[3] N. Collins and J. d'Escrivan, Eds., *The Cambridge Companion to Electronic Music*. Cambridge University Press, 2007.

[4] G. T. Toussaint, "The Geometry of Musical Rhythm." in *Lecture Notes in Computer Science*, J. Akiyama, M. Kano, and X. Tan, Eds., vol. 3742. Springer, 2004, pp. 198–212.

[5] G. Toussaint, "Classification and Phylogenetic Analysis of African Ternary Rhythm Timelines," in *Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science*, 2003, pp. 25–36.

[6] A. Allik, "Gene expression synthesis," in *Proceedings of the ICMC/SMC, Athens, 14-19 September*, 2014.

[7] C. Ferreira, "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems," *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001.

[8] D. Mitrovic, M. Zeppelzauer, and C. Breiteneder, "Features for Content-Based Audio Retrieval," *Advances in Computers*, vol. 78, 2010.

[9] M. Lanthaler and C. Gütl, "On Using JSON-LD to Create Evolvable RESTful Services," in *Proceedings of the 3rd International Workshop on RESTful Design at WWW2012*, 2012.

[10] R. Phillips and E. W. Weisstein, "Outer-Totalistic Cellular Automaton." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Outer-TotalisticCellularAutomaton.html.

[11] M. Wojtowicz, "Mirek's Cellebration - 1-D and 2-D Cellular Automata viewer, explorer and editor," http://www.mirekw.com/ca/index.html, accessed: 2012-10-15.

[12] T. Kohonen, *Self-Organising Maps*. Springer-Verlag, 1994.

A Differential Equation Based Approach to Sound Synthesis and Sequencing

B. A. Jacobs
School of Computer Science and Applied Mathematics,
DST-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS),
University of the Witswatersrand, Johannesburg,
Private Bag 3, Wits 2050, South Africa
byron.jacobs@wits.ac.za

ABSTRACT

Differential equations have been extensively used to model dynamical systems and provide sophisticated tools for their analysis. This work explores the potential of applying differential equations to the musical synthesis of sound and also a rudimentary algorithmic composition tool through transport equations. The Fitzhugh-Nagumo relaxation oscillator is implemented as a sound generator and is found to exhibit a rich harmonic frequency spectrum. The sequencing of tones is done through the modelling of impulses propagating on a randomly generated network, inspired by star constellations, governed by a transport equation. The propagation of different initial conditions on the network define an amplitude envelope for the excitation of each node on the graph.

1. INTRODUCTION

The study of the application of mathematics to sound design, synthesis and musical applications has been underway for many years. However the use of differential equations has been relatively under utilised in this field. Recently Stefanakis et al. [1] presented a robust framework for the synthesis of sounds by coupling together real and complex valued ODEs. One component of this system acts as an envelope for the sound where the other the other component determines the frequency of an oscillator. Through nonlinear interactions between these components one may synthesize sounds whose frequencies depend on the amplitude envelope of the sound itself, as in the case of "tension modulation" in a plucked string. The spectral composition of these sounds may be enriched by the superposition of several oscillators with harmonic frequencies of the fundamental frequency.

The application of mathematics in capturing musical and sonic characteristics has been deeply investigated by many authors. Slater [2] explores the relationship between the chaotic behaviour of ODEs and frequency-modulation (FM) synthesis. FM synthesis, popularized in the 1980s by Yamaha with their DX synthesizers, is achieved by modulating the frequency of one oscillator with the frequency of another oscillator typically with audio rate oscillations. This is captured in the dynamic ODE models as a nonlinearity which often results in chaotic behaviour for certain parameter choices. Sonically this results in musically interesting behaviour since in chaotic systems small changes in parameters or initial conditions result in an unpredictable trajectory in phase space. The construction and analysis of Chua's circuit for real-time musical performance is conducted by Choi [3] where the circuit exhibits chaotic behaviour lending itself to interesting musical opportunities. The idea of synthesizing sounds from a set of instructions as opposed to the imitation of an existing sound is termed "non-standard" and is succinctly discussed by Döbereiner [4].

There is of course an enormous effort toward the digital synthesis of physical instruments. Röbel [5] illustrate the efficacy of using dynamical strange attractors to synthesize natural sounds, including that of the saxophone, flute and piano. Bilbao [6] presents a thorough overview of numerical techniques toward the real-time and accurate synthesis of physical models. Some intriguing numerical techniques have arisen from this work including inharmonic percussion synthesis by Torin et al. [7] through an energy conserving finite difference scheme. Another approach to physical modelling is taken by Rabenstein and Trautmann [8] using the functional transform method for the real-time realisation of physical models.

Algorithmic composition is another aspect of music in which mathematics and computation has been applied. A review of techniques for algorithmic composition is conducted by Supper [9]. Nierhaus [10] also provides a detailed overview of algorithmic composition techniques as well as a historic recollection. An analytic framework for the evaluation of existing methods for algorithmic composition is presented in [11] by Leach and Fitch.

Weinberg [12] discusses the aesthetic and technical impacts of technology on networks of music generation which allow parameters of one performer to influence the performance of others on the network.

Some work has even been done in borrowing from musical ideas to construct new mathematical ideas. Alatas [13] presents an optimization method based on musically inspired heuristics which exploit chaotic maps for parameter estimation which allow the scheme to avoid the erroneous convergence on local optima.

The exploratory work conducted in this paper presents some of the opportunities for sound design, synthesis and sequencing through the construction of a system of ordinary differential equations (ODEs). The approach comprises three main components, a network of nodes connected in a graph structure, a system of transport equations which describe the propagation effects of impulses throughout the network and finally each node in the network represents an oscillator, the sound of which is synthesized through the solution of a system of ODEs, known as the Fitzhugh-Nagumo relaxation oscillator.

A fundamental frequency is assigned to each node in the graph network which can represent an atonal collection of frequencies, chromatic scale or otherwise. The desired musical content is at the discretion of the user and the framework provided is robust to accommodate this. As each node in the graph is excited by the propagating envelope signal the node will generate its corresponding tone given its fundamental frequency. The spectral content of this tone is dependent on the parameters of the model being solved.

The following section, Section 2, outlines the methodology for each component of the proposed approach, including the numerical methods used and justifications for their use. Section 3 presents the results obtained by the proposed method together with some spectral analysis. Finally some concluding remarks and possible extensions are discussed in Section 4.

2. METHODOLOGY

This section presents the methodology which was used to construct the synthesis framework described above. Figure 1 presents a high-level representation of the algorithmic procedure.

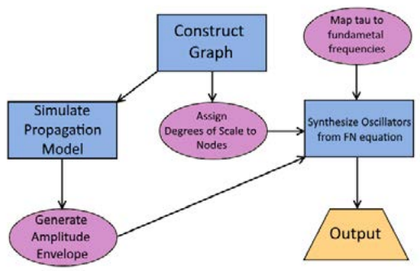


Figure 1. Flow Chart Illustrating Algorithmic Procedure.

2.1 Graph Construction

Initially a random directed graph is constructed, this idea was conceived as a representation of a constellation of stars in keeping with the theme of the 2016 ICMC. Cycles in the graph correspond to a recurring pattern of notes, while the edge weights of the graph can be interpreted as a length. For example if an edge has length two a wave propagating at unity will take twice as long to reach the adjacent node as a edge of weight one. Musically this introduces a sense of rhythm as well as potential motifs in the case of cycles. We restrict the edge weight to be powers of two (both negative and positive) to restrict the interval between exciting nodes to musical note values (eighth note, quarter note, half note and whole note).

2.2 Propagation Model

The propagation of impulses between nodes on the graph is governed by a transport equation written in conserved form as

$$\mathbf{u}_t = f(\mathbf{u})_x, \tag{1}$$

where the subscripts denote the partial derivative and $\mathbf{u}(x,t) = \{u_1(x,t), u_2(x,t), \dots u_N(x,t)\}$, where N is the number of nodes in the graph. The network of equations are coupled together by their boundary conditions, $u_q(0,t) = u_p(1,t)$ for all (p,q) pairs such that node p is directed to node q in the prescribed graph.

The choice of $f(u)$ determines the nature of the propagation dynamics of the model and ultimately affect the amplitude envelope for the resulting sound. For simplicity we take $f(u) = u$ which results in the linear advection equation describing the motion of the envelope. The linear advection equation preserves the shape of the initial condition which provides a uniform envelope to each node as they are excited. Another possibility is to take $f(u) = 1/2u^2$ which would result in the inviscid Burgers' equation which exhibits a steepening in the gradient of the wave front. This would correspond to a fast attack in the envelope with some decay.

In order to numerically simulate this transport equation we implement a Lax-Wendroff finite difference scheme [14]. This numerical scheme mitigates the effects of numerical diffusion and provides a stable scheme for the simulation of the model. Each transport variable in $\mathbf{u}$ is discretized as $u_k(x_i, t_n) \approx {}_k u_i^n$ where $k = 1, 2, \dots, N$. The spatial discretization is given by $i = 0, 1, \dots, M_k$ so that $M_k \Delta x = L_k$ where L_k is the 'length' of the graph edge along which the envelope is propagating.

The numerical scheme approximating equation (1) is given by

$${}_k u_i^{n+1} = {}_k u_i^n - \frac{\Delta t}{\Delta x}\left(f({}_k u_{i+1/2}^{n+1/2}) - f({}_k u_{i-1/2}^{n+1/2})\right), \tag{2}$$

where

$${}_k u_{i+1/2}^{n+1/2} = \frac{1}{2}\left({}_k u_{i+1}^n + {}_k u_i^n\right) - \frac{\Delta t}{2\Delta x}\left(f({}_k u_{i+1}^n) - f({}_k u_i^n)\right), \tag{3}$$

and

$${}_k u_{i-1/2}^{n+1/2} = \frac{1}{2}\left({}_k u_i^n + {}_k u_{i-1}^n\right) - \frac{\Delta t}{2\Delta x}\left(f({}_k u_i^n) - f({}_k u_{i-1}^n)\right). \tag{4}$$

A possible initial condition is given by

$$\begin{aligned} {}_1 u(x,0) &= e^{-25x^2}, \\ {}_2 u(x,0) &= 0, \\ &\vdots \\ {}_N u(x,0) &= 0, \end{aligned} \tag{5}$$

(6)

which produces an amplitude envelope with soft attack and decay and no sustain at the first node and not exciting any other nodes initially. The choice of initial condition can be tailored based on the desired amplitude envelope. As previously mentioned the transport equations are coupled together through their boundary conditions and are numerically imposed as

$${}_q u_0^n = {}_p u_{M_k}^n, \tag{7}$$

for all (p,q) pairs such that node p is directed to node q in the prescribed graph, as before.

2.3 Synthesis Model

The Fitzhugh-Nagumo [15, 16] model for the impulse propagation along a neuron axon is known to admit oscillatory solutions and is an example of a relaxation oscillator. The primary motivation for using this model to generate the oscillations is that the solution admits a spectrally rich harmonic waveform. Traditionally these spectrally rich waveforms can be constructed by taking a superposition of sinusoids at harmonic frequencies with various weightings, however the solution to the Fitzhugh-Nagumo model elegantly admits an interesting waveform. Moreover this waveform can be tweaked through the model parameters in interesting ways to produce new waveforms. The model is given by

$$\frac{dv}{dt} = v(t) - \frac{v(t)^3}{3} - w(t) + I_{\text{ext}}, \tag{8}$$

$$\tau\frac{dw}{dt} = v(t) + a - bw(t), \tag{9}$$

with $w(0) = 0$ and $v(0) = 0$. The system is then excited through the I_{ext} parameter, to avoid the system being self excited we take $a = 0$. The system also becomes locked in a non-oscillatory state for large values of b and hence we take $b = 0$ as well, although varying b admits some minor changes in timbre and pitch. The first equation in v represents what's known as the fast variable and is a voltage like oscillator that exhibits self exciting behaviour through positive feedback, where as the slow variable w with linear dynamics exhibits a negative feedback. The variable τ scales the excitation rate of the fast variable $v(t)$ and is used to tune the oscillator to the required fundamental frequency. By examining the Fourier decomposition of the signal produced by this equation we see that the spectral content is comprised of odd harmonics of the fundamental frequencies. Figure 2 illustrates the frequency spectrum of the Fitzhugh-Nagumo oscillator, up to 11000Hz, with a fundamental frequency of 440Hz. The waveforms generated by the Fitzhugh-Nagumo oscillator are illustrated in Figure 3, noting that the solution $v(t)$ yields a waveform similar to an asymmetrically softened square wave, where as the $w(t)$ solution admits a triangular type waveform.

Tuning of the oscillator to a specific fundamental frequency can be achieved through the rescaling parameter τ. By assuming there exists a smooth underlying inverse relationship between τ and the fundamental frequency, i.e. larger values of τ admit lower fundamental frequencies and vice versa, a cubic interpolation of constructed data provides an efficient method of generating τ values for a prescribed fundamental frequency. This relationship is presented in Figure 4.

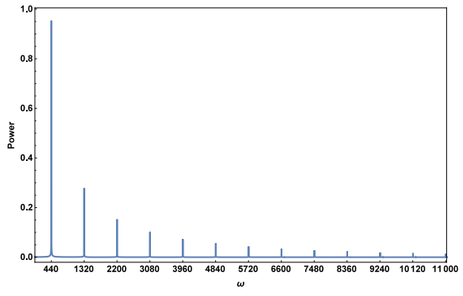


Figure 2. Frequency Spectrum of the Fitzhugh-Nagumo Relaxation Oscillator with a Fundamental Frequency of 440Hz.

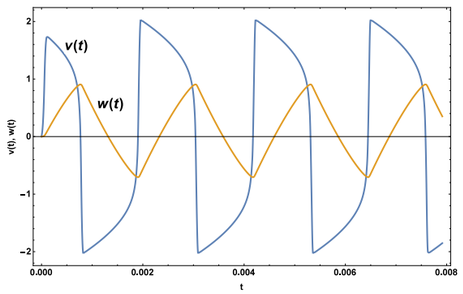


Figure 3. Time Series of the Fitzhugh-Nagumo Relaxation Oscillator with a Fundamental Frequency of 440Hz.

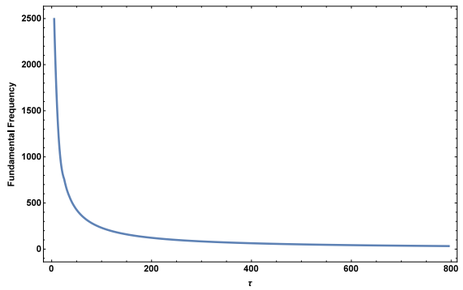


Figure 4. An Interpolating Function Capturing the relationship between τ and the Fundamental Frequency of the Fitzhugh-Nagumo Oscillator.

3. RESULTS

3.1 Example 1: Eight Node Cyclic Graph

This example examines the graph with eight nodes corresponding to the seven degrees of the Ionian scale and an octave of the root note, illustrated in Figure 5. Defining the

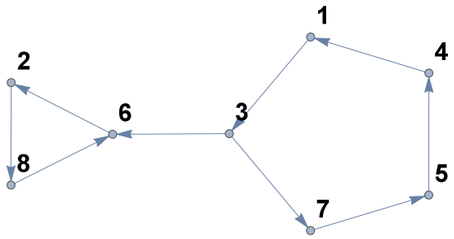


Figure 5. Graph depicting connected degrees of the A Ionian Scale.

initial condition as $u_1(x,0) = \max(1-|4x|,0)$ provides a triangular envelope. This envelope propagates along the network according to the system of transport equations, the time domain excitation along the nodes is illustrated in Figure 6, where each colour corresponds to a different node on the graph. The composition admitted by one realization of

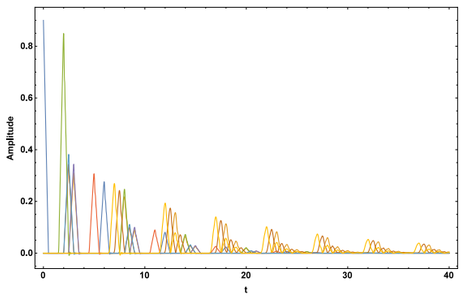


Figure 6. Amplitude Envelope Resulting from the Linear Advection Equation.

this simulation is atypical, rhythmically interesting as well compositionally unusual, where degrees of the chords are constructed randomly but cycles within the graph provide recurring patterns. The spectrogram of the resulting synthesis is given in Figure 7.

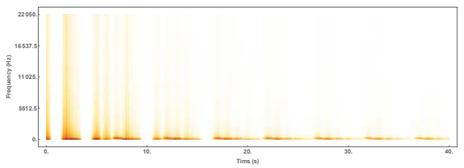


Figure 7. Spectrogram of Synthesis Result from Example 1.

3.2 Example 2: Twelve Node Cyclic Graph

A graph with twelve nodes corresponding to the chromatic scale is constructed, shown in Figure 8. Defining the initial

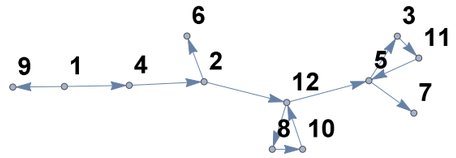


Figure 8. Graph depicting connected degrees of the Chromatic Scale.

condition as $u_1(x,0) = e^{-25x^2}$ defines a smooth Gaussian envelope. This envelope propagates along the network according to the system of transport equations as illustrated in Figure 9, again where different colours correspond to different nodes on the graph. By imposing an exponen-

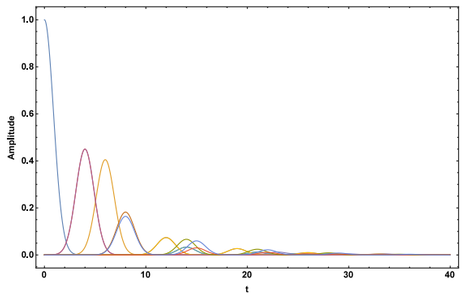


Figure 9. Amplitude Envelope Resulting from the Linear Advection Equation.

tial initial condition the amplitude envelope experiences a slower attack and decay and as such produces soundscape that has a subtle rhythmic character. The selection of a chromatic scale also introduces unusual 'accidentals' into the composition. The spectrogram in Figure 10 illustrates the smooth transient sounds generated in this example due to the Gaussian initial condition.

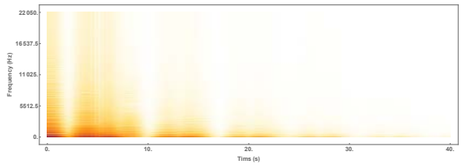


Figure 10. Spectrogram of Synthesis Result from Example 2.

4. CONCLUSION

We have illustrated how a physically based framework may be used unconventionally as a method of not only sound synthesis but also a rudimentary algorithmic composition technique. Networks describing musical ideas may be generated and the physical parameters of the proposed models can be nearly infinitely altered to generate new and interesting musical ideas.

A continuation of this work into the analysis of the Fitzhugh-Nagumo equation could potentially yield an analytic relationship between the τ parameter and the fundamental frequency of oscillation, this would replace the need to 'tune' the oscillator.

The Fitzhugh-Nagumo equation was selected due to the variety of unique and sonically rich waveforms which may be generated from this equation through the variation of the model parameters. The dynamic nature of differential equations allows cross coupling of parameters between transport model and the synthesis model, as one would do when patching components of a modular synthesizer together. Furthermore there are several aspects of sonification, over and above pitch and amplitude, that may be controlled, for example spatial location (stereo placement of sound), phase, rates of change of parameters, etc.

The method introduced by Stefanakis [1] could be implemented for the synthesis of sounds (where the amplitude envelope is prescribed by the solution of the coupled equation) while the method of impulses propagating on the network presented here being used purely as a sequencing mechanism. Counter to this, part of the present method's appeal is the dynamic nature of the transport model and how it affects the evolution of the amplitude envelope, which raises interesting questions relating to how different models can be used for different results.

The transport model can also be extended to include diffusive properties, this would guarantee convergence of the excitation in the linear case. Sonically, diffusion would introduce a softening of the amplitude envelope, which may or may not be desirable. The shape of the amplitude envelope is affected by both the initial condition imposed on the system as well as the model itself, as alluded to by the consideration of the Burgers' equation which would experience a gradient steepening as the wavefront propagated along the network edges.

This work has illustrated one aspect of mathematical modelling of physical phenomena as applied to the synthesis and sequencing of musical composition. This framework allows for a vastly dynamic and novel approach to sonic exploration, while still providing an analytic background from which deep analysis into the behaviour of the system may be conducted. This understanding of the underlying physical processes provides control over the final composition.

Acknowledgments

The author would like to thank the NRF for their support under Thuthuka grant (94005) and the DST-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS). Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the CoE-MaSS.

5. REFERENCES

[1] N. Stefanakis, M. Abel, and A. Bergner, "Sound synthesis based on ordinary differential equations," *Computer Music Journal*, 2015.

[2] D. Slater, "Chaotic sound synthesis," *Computer Music Journal*, vol. 22, no. 2, pp. 12–19, 1998.

[3] I. Choi, "Interactive exploration of a chaotic oscillator for generating musical signals in real-time concert performance," *Journal of the Franklin Institute*, vol. 331, no. 6, pp. 785–818, 1994.

[4] L. Döbereiner, "Models of constructed sound: Non-standard synthesis as an aesthetic perspective," *Computer Music Journal*, vol. 35, no. 3, pp. 28–39, 2011.

[5] A. Röbel, "Synthesizing natural sounds using dynamic models of sound attractors," *Computer Music Journal*, vol. 25, no. 2, pp. 46–61, 2001.

[6] S. Bilbao, *Numerical Sound Synthesis*. Wiley Online Library, 2009.

[7] A. Torin, B. Hamilton, and S. Bilbao, "An energy conserving finite difference scheme for the simulation of collisions in snare drums." in *DAFx*, 2014, pp. 145–152.

[8] R. Rabenstein and L. Trautmann, "Digital sound synthesis of string instruments with the functional transformation method," *Signal Processing*, vol. 83, no. 8, pp. 1673–1688, 2003.

[9] M. Supper, "A few remarks on algorithmic composition," *Computer Music Journal*, vol. 25, no. 1, pp. 48–53, 2001.

[10] G. Nierhaus, *Algorithmic composition: paradigms of automated music generation*. Springer Science & Business Media, 2009.

[11] J. Leach and J. Fitch, "Nature, music, and algorithmic composition," *Computer Music Journal*, vol. 19, no. 2, pp. 23–33, 1995.

[12] G. Weinberg, "Interconnected musical networks: Toward a theoretical framework," *Computer Music Journal*, vol. 29, no. 2, pp. 23–39, 2005.

[13] B. Alatas, "Chaotic harmony search algorithms," *Applied Mathematics and Computation*, vol. 216, no. 9, pp. 2687–2699, 2010.

[14] P. Lax and B. Wendroff, "Systems of conservation laws," *Communications on Pure and Applied mathematics*, vol. 13, no. 2, pp. 217–237, 1960.

[15] R. FitzHugh, "Mathematical models of threshold phenomena in the nerve membrane," *The bulletin of mathematical biophysics*, vol. 17, no. 4, pp. 257–278, 1955.

[16] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon," *Proceedings of the IRE*, vol. 50, no. 10, pp. 2061–2070, 1962.

Music Poet: A Performance-Driven Composing System

Li Rongfeng
Beijing University of Posts and Telecommunications
lirongfeng@bupt.edu.cn

Zhang Xinyun
Peking University
xinyunzhang@pku.edu.cn

Bi Minghui
Peking University
biminghui@pku.edu.cn

ABSTRACT

Innovation is more and more difficult in conventional music composing as we enter the new century. In the past century, gaining inspiration from other forms of art is a main innovative composing method. However, to compose music by other forms of art, composers have to be professional in both music and that art form.

In our work, we developed a performance-driven composing system named Music Poet. Based on the features of the poem and the music pieces in the database, the computer can automatically compose a piece of music. On one hand, professional musicians who do not know much about poetry can get music material by transforming poem into music. On the other hand, amateur composers who have not enough music knowledge but know quite well about poetry can also create original music by composing a poem instead of music. Thus, the composers' creativity will be no longer limited by the human's profession. We believe that the new music material, new methods of information acquisition and data using, and new interactive way can break the limit of traditional composing method and will stimulate composers' creativity.

1. INTRODUCTION

Innovation is more and more difficult in music composing as we enter the new century. Composers have made great effort looking for new music material and developing new composing techniques during the past century.

In the 20th century, in order to get innovative music material, many composers adopted expressionistic style. Based on the idea of expressionism, originality and personal expression have become vital to artists, and artistic work is required to express and reveal human's inner feelings and spirits. Composers also gain inspiration from the performance of other forms of expressionistic art like painting, dancing or other visual arts.

The exchange between musicians and painters climaxed in the middle of the 20th century, when a group New York School [1], including John Cage and Morton Feldman formed friendship with a group of painters. Morton Feldman's "Why Patterns" (the music score of which is shown in Figure 1) is composed according to the texture principles derived from Jasper John's crosshatch paintings (shown in Figure 2).

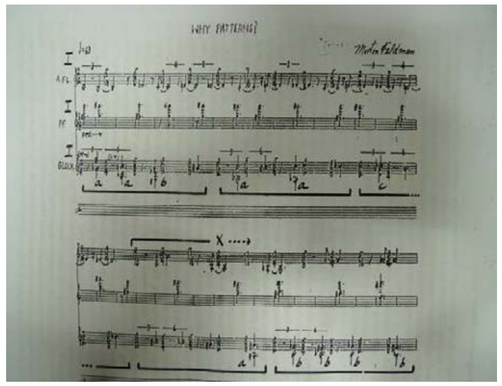


Figure 1. Music Score of Feldman's "Why Patterns"

Figure 2. John's Crosshatch Painting

We call this kind of work "performance-driven" composing. The "performance-driven" method is well known by a few composers nowadays. The method involves the professional knowledge of other forms of art. Thus, not only does the composer need to be a musician, but also they have to obtain the expertise of painting, dancing and other forms of arts.

Making use of the intelligent technologies of computer music, pattern recognition, machine learning and human-computer interaction, database and internet, computer can offer some help in "performance-driven" composing. With the pattern recognition technique and some experience of the composers, computers can detect the features from image, voice, video, text, etc. and learn the relationship between the music feature and the features from other art performance. This makes it possible to transform other forms of art into music automatically. In this way, composers who do not have enough musical knowledge but have some knowledge of other arts, can get innovative music material. Thus, the composers' creativity will be no longer limited by the one or two composers' knowledge. The online music database is updated by the users of this system itself. In this way, composers can share composing experience and music information with their music in a new way rather than the traditional communications.

In our work, we proposed an innovative computer-assisted way of composing. Here, our goal is innovative composing methodology but not the performance form (which is still traditional way including concert, CDs and internet music, etc.). We will implement a performance-driven composing system, namely Music Poet. Composers compose poet instead of music. Music Poet will automatically transform the poet into music. Music Poet provides an interactive platform to everyone even who do not know much about music, on which they can work with computer to share music with each other and compose. We believe that the new music material, new methods of information acquisition and data using, and new interactive way can break the limit of traditional composing method and will stimulate composers' creativity.

2. PERFORMANCE-DRIVEN COMPOSING SYSTEM

Traditional composing activity is based on the following procedure shown in Figure 3.

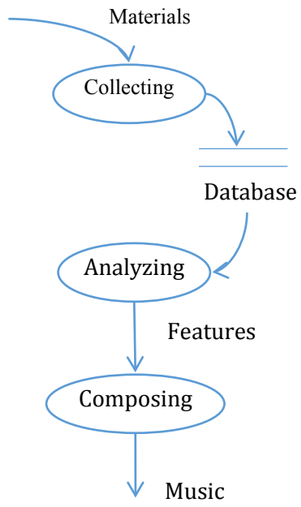


Figure 3 Traditional Composing Activity

In this procedure, composers will first collect music materials (such as folk songs and instrumental music pieces) and then get the data (recording and musical score). After that, they analyze the data and extract some interesting features (pitch form, rhythm pattern, etc.). Finally comes the work of composing. We call this activity materials-driven. Nowadays, it is very difficult to get totally innovative music material in the form of sound, because musicians have already tried almost all possible kinds of sound, noise and even silence (The extreme example is John Cage's 4'33'', the score of which instructs the performer not to play the instrument during the entire duration of the piece throughout the three movements) in the past century.

Inspired by the modern musicology, we can look at the composing procedure in a new way, which is shown by the following procedure in Figure 4.

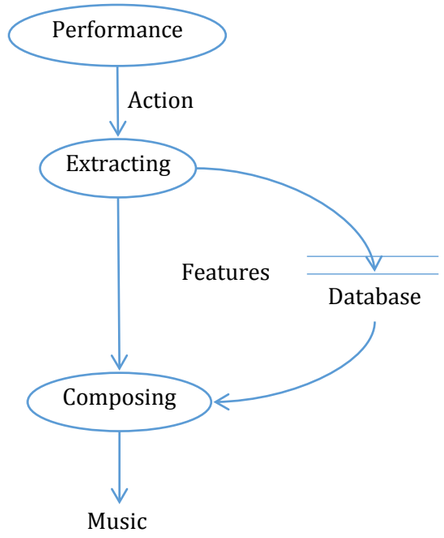


Figure 4. Performance-Driven Composing Activity

In the performance-driven composing procedure, we will get the source materials directly from the user's action. They are allowed to express themselves in any possible art form. A user may read aloud a poem, draw a picture, or perform a dance. Computer captures the actions and extracts the features. Then, based on the matching function (generated by Machine Learning approaches) between the features and music, the system retrieves the matched music pieces from the music database. Finally, the system automatically composes with the features of actions and the retrieved music pieces.

3. MUSIC POET

Based on the composing procedure above, we developed an automatically composing system named Music Poet. Music Poet is an application based on the research of relationship between Chinese ancient poem and the corresponding music score *Gongche-Pu* ([2] [3] [4]). Traditional Chinese Musicians composed the melody based on the tone of each word in the poem. We will introduce some statistical models to do the automatically composing using the database of *Gongche-Pu*.

3.1 Chinese Traditional Music Score *Gongche-Pu*

Chinese poetic songs are noted by Gongchepu-Chinese traditional musical notation, once popular in ancient China and still used for traditional Chinese musical instruments and Chinese operas nowadays. A Gongchepu sample of Chinese poetic songs entitled 天□沙 Tian-jin-sha is shown in Figure 5.

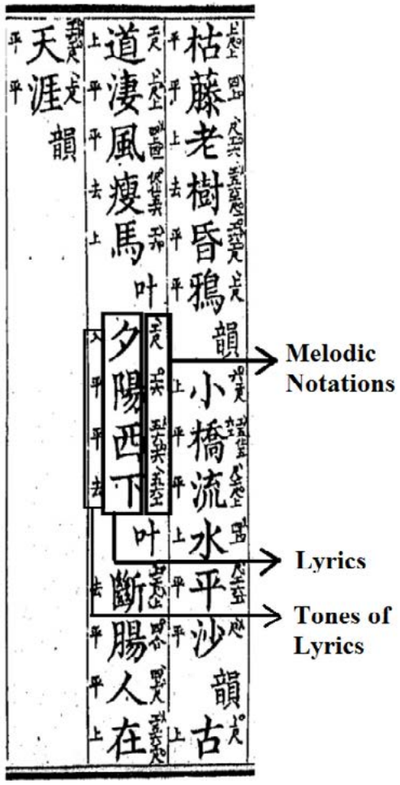


Figure 5. Gongchepu of Tian-jin-sha

As illustrated in Figure 5, the tone of each word is noted by the left side of lyrics. The melodic notations of *Gongche-Pu* are noted at the right side of the lyrics, consisted of pitch notation and rhythm notations, which are the two basic characters of a musical notation.

Pitch Notation

Pitch of each note in Gongchepu is denoted by 10 Chinese characters:合 hé ,四 sì, 一 yī, 上 shàng, 尺 chě, 工 gōng, 凡 fán, 六 liù, 五 wǔ, 乙 yǐ. They are equivalent to the notes of solfège system: sol, la, ti, do, re, mi, fa, sol, la, ti. 合 hé ,四 sì, 一 yī are pitched an octave lower 六 liù, 五 wǔ, 乙 yǐ. Gongchepu is named by the character 工 gōng and 尺 chě.

Once we take 上 shàng as the fixed pitch c1, the range of the 10 characters is g-b1.

Gongchepu uses the following notations to note other notes in different octaves:

a) Octaves higher: a radical "亻" is added for one octave higher. For example, we use "□" to represent an octave higher "上". Similarly, the radical "彳" is added to represent two octaves higher.

b) Octaves lower: an attached stroke is added to the ending of stroke of the character to note an octave lower. For example, we use "∨" to show an octave lower "上".

Likely, two attached parts are added to represent two octaves lower.

Rhythm Notation

Now we will explain the rhythmic rules of Gongchepu. Gongchepu denotes the beats by the following notations: The mark "、" represents the stronger-beat which is called ban, while the notation "。" represents the off-beat called yan. The marks are put at the upper right corner of the first note of a beat. Illustrated from Figure 6 which is written horizontally for convenient reading, we can see the notes separated into beats with the ban and yan.

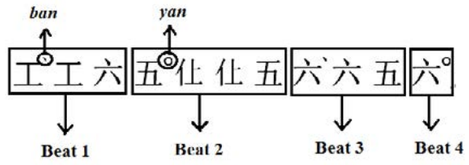


Figure 6. Ban and yan in Gonchepu

Rhythmic structure of gongchepu is formed by the regular combination of ban and yan. For example, the cycle of 1 ban and 1 yan forms a 2/4 mater and cycle of 1 ban and 3 yan forms a 4/4 mater. However, the duration of each note, which should be noted in staff, cannot be specified by the rhythmic mark of ban and yan. In this case, the rhythm notations cannot be interpreted to the exclusive corresponding notations. For example, if 2 notes are in 1 beat, it can be sung as , or . If 3 notes are in 1 beat, we could get 4 results: , , and .

3.2 Generate Pitch from the Tone of Poem

Chinese language consists of 字 Zi (Chinese character) which is a word unit with a single ideograph, pronounced with a single speech- tone, roughly equivalent to one syllable. Although the pronunciations of the Zi in *Gongche-Pu* have an origin lost in the mists of antiquity, the four type tones of Zi are noted in Gonghepu. They are 平 ping (level-tone) , 上 shang (rising-tone), 去 qu (falling-tone) and 入 ru (entering-tone). The pitch-time features of the four tones can be illustrated in Figure 7.

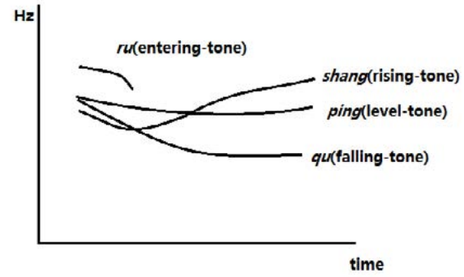


Figure 7. Pitch-time features of the four tones

There is a common view that the melody of Chinese poetic songs is evolved from the tones of Zi.

We indicate the tone of each Zi in Gongche-Pu by T1, T2, …, Tn for a poem and encode the four tone by ping-1, shang-2, qu-3 and ru-4. For example, The tone of *夕阳西下* from Figure 3 is 入 ru 平 ping 平 ping 去 qu, thus the sequence {T1,T2,T3,T4}={4,1,1,3}.

We indicate the Pitch of each Zi by P1,P2,…,Pn and encode by the following "interval direction and position" which is introduced by Williams [5] for melodic analysis. Williams use "+" for rising direction of the pitch interval and "-" for the falling direction. Moreover, pitch interval is measured by chromatic scale. For example, the pitch interval direction and position of the section of "Tune of Fresh Flowers" is illustrated in Figure 8.

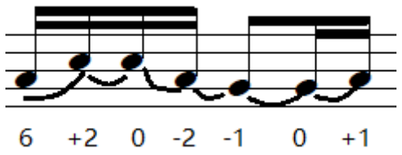


Figure 8. Pitch interval direction and position

For example, the pitch of 夕 of Figure 3. is 工 mi 尺 re, then P1={3,-1}.

Thus, the relationship between tone and pitch can be describe by the sequence model by T1,T2,…,Tn and P1,P2,…Pn which is shown by Figure 9.

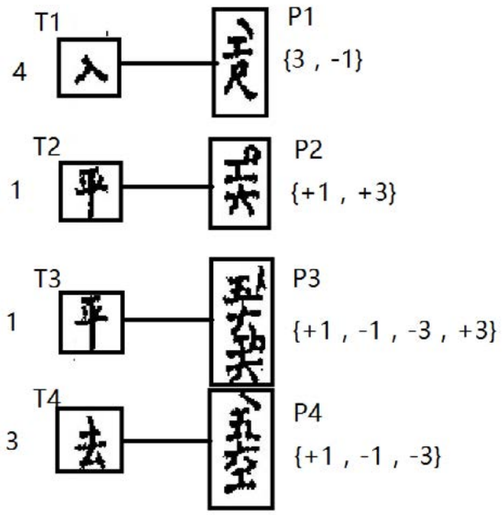


Figure 9. Sequence Model of Tone and Pitch

Statistical model such as Hidden Markov Model, Conditional Random Field can be applied to generate the pitch from the new tone sequence, taking the Gongche-Pu database as training database.

3.3 Generating Rhythm from the Pitch

Once we finish generate the pitch, the rhythm generation problem will be the same as a rhythm interpretation of [2]. We denote the beat sequence by B1,B2,…,Bn Taking the "Tune of Fresh Flowers" as an example, beats separations are shown in Figure 10.

Figure 10. Beat separation by marks of ban and yan

However, the duration of each note, which should be noted in staff, cannot be specified by the rhythmic mark of ban and yan. In this case, the rhythm notations cannot be interpreted to the exclusive corresponding notations. For example, if 2 notes are in 1 beat, it can be sung as , or . We indicate the rhythm pattern of each beat by R1, R2, … , Rn.

Interpret the notes beat by beat, the interpretation task is illustrated in Figure 11.

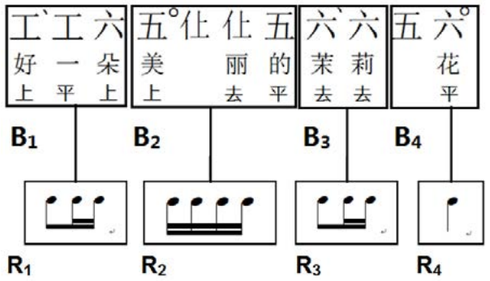


Figure 11. Interpret the rhythm beat by beat

In spite of the missing information of the duration of each note, the length of note duration in a beat is relatively fixed. Thus, rhythm patterns of each beat are limited. In this paper, we conclude 37 patterns p1, p2,…,p37 which are used in Chinese poetic music. Thus, the value of Ri, i=1, 2, …, n is limited in the patterns set S={p1, p2,…,p37}.

By the above denotations, the interpretation transform to a tagging problem: when the beats sequence {B1,B2,…,Bn} is observed, we are required to tag the sequence by the rhythm patterns from a limited set P. This is very similar to the sequence tagging problem in natural language processing.

Once the features F(Bi)={f1(Bi), f2(Bi),.., fm(Bi)} of each beat are extracted, statistical language processing models can be applied to the interpretation based on different assumptions.

4. EXPERIMENTAL RESULTS

The Gongche-Pu database is based on The Complete Works of Jiu-Gong [6] which is compiling by Qin Yun Wang in Qing Dynasty. The Complete Works of Jiu-Gong collected 4466 Chinese Poem of Tang, Song Yuan, Ming and Qing Dynasty with the corresponding musical score (in the form of Gongche-Pu). We use the digital version of the musical score in Jiu-Gong to set up the statistical model of pitch generation in 3.2 and use the program of [2] to generate the rhythm.

We take Chinese Poem Huan-Xi-Sha as an example. The input of the Music Poet is shown in Figure 12. Tones are already noted on the left side of the lyrics.

Figure 12 shows the result of the Muic Poem.

Since there is no definitely right or wrong in music composing, we can not quantitatively evaluate our system. Instead, we perform the music in the concert and get the comment from the audience. The fantasia “Gu Feng” is composed by Li Bochan by the help of Music Poet. The music is played in the Edinburgh Music Festival by PKUCMICO (Peking University Chinese Music Institute Chinese Orchestra). The following link is the video of Gu Feng. https://www.youtube.com/watch?v=g-4p6o7Rh8Y

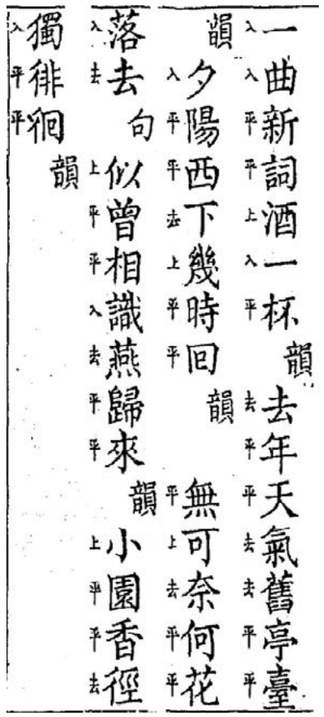


Figure 11. Huan-Xi-Sha 浣溪沙 as an input of Music Poem

浣溪沙

Figure 12. The result of Music Poem

5. CONCLUSIONS

In this paper, we introduced a performance-driven composing system. Taking poetry as an performance form, we shown Music Poem as an implement of the composing system.

The performance-driven composing system is very useful for the professional composers. Composers can transform other forms of art such as poems into music and don’t have to be professional on that. Once we have the database which can be used to train the statistical model of the relationship between music and other art form, we can obtain plenty of new music from the database and without the limited of human’s profession.

Music Poem also benefit the amateur composers who don’t have enough music knowledge but have some knowledge of other art. Once they complete a poem, the corresponding music of the poem is composed automatically and even can be sang with the lyrics of the poem.

The evaluation of the composing system is still a hard work because there is no right or wrong in music. In further research, we will develop an online version of Music Poet in order to collect the feedback of users and evaluate the system by scoring and comment

6. ACKNOWLEDGE

This work is supported by Youth Innovation Project of BUPT under Grant No.2015RC32

7. REFERENCES

[1] Steven Johnson, “The New York Schools of Music and the Visual Arts”, Routledge, New York, November 7, 2001, ISBN-10: 0415936942

[2] Rongfeng Li, Yelei Ding, Wenxin Li and Minghui Bi, “Automatically Interpretation of Chinese Traditional Musical Notation Using Conditional Random Field”, International Symposium on Computer Music Modeling and Retrieval (CMMR) ,London, UK, June 19-22 2012

[3] Rongfeng Li, Yelei Ding, Wenxin Li and Minghui Bi, “Literarily Dependent Chinese Music: A Cross-Culture Research of Chinese and Western Musical Score Based on Automatic Interpretation”, International Conference on Music Perception and Cognition (ICMPC) ,Thessaloniki, Greece, July 23-28 2012

[4] Rongfeng Li, Yelei Ding, Wenxin Li and Minghui Bi, “A Classification Approach to Interpretation of Traditional Chinese Musical Score”, International Conference on Information Science and Technology (ICIST), Wuhan, China, March 23-25, 2012

[5] J . Kent Williams, “Theories and Analyses of Twentieth—century Music”, Harcourt Brace & Company.1997.

[6] Qin Yun Wang,“The Complete Works of Jiu-Gong”, 1746.

Multiple Single-Dimension Mappings of the Henon Attractor as a Compositional Algorithm

Alexandra Kurepa
NC A&T State University
kurepa@ncat.edu

Rodney Waschka II
North Carolina State University
waschka@ncsu.edu

ABSTRACT

This paper describes different uses of a compositional algorithm based on the Henon Map in the creation of a new art-music composition, Au Revoir, Svetozar*, for string quartet and guzheng, composed by author Waschka. Unlike other utilizations, the musical mappings here employ the output as single-dimensional data, rather than two-dimensional data and include the use of the Henon Attractor to control a non-isorhythmic cantus firmus and to specify pitch material for the other parts. The Henon Map is a discrete, deterministic dynamical system that exhibits chaotic behavior depending the setting of two parameters –* a *and* b*. The mappings described here feature different settings for these two parameters depending on the instruments and musical usage. The initial conditions employed, the resulting output, and the mapping to the musical parameters are shown along with excerpts from the resulting music composition*. Au Revoir, Svetozar *received its premiere performance, given by the Hong Kong New Music Ensemble, at the Echofluxx Festival in May 2016 in Prague.*

1. INTRODUCTION

The Henon Map and the Henon Attractor are named for the French astronomer, Michel Henon, who introduced the system in a seminal article [1]. The Henon Map is a two-dimensional deterministic dynamical system defined by the equations:

$$x_{(n+1)} = 1 - a(x_n)^2 + y_n \quad (1)$$

$$y_{(n+1)} = bx_n \quad (2)$$

The system depends on two parameters -- *a* and *b*. One set of parameters known to produce chaotic results are *a* = 1.4 and *b* =0.3. These particular values are often used to produce the typical Henon Attractor.

The values selected for *a* and *b* determine whether the results produced through iteration will move towards a stable (repeating) orbit or will display chaotic (non-repeating and unpredictable) behavior. M.J. Islam et al have documented the regions of stable and chaotic behavior [2].

Various ways of employing chaotic dynamical systems as the basis for music composition algorithms have been described by Jeff Pressing [3], Waschka [4] and others. The Henon Map has been employed to produce a string of *x* and *y* data points that can be used for musical demonstration purposes [5] or to generate melodies, as in Clark [6] and Coca et al [7]. Work specifically using the Henon Map for art-music composition has been described by Waschka and Kurepa and others [8].

Often, these researchers have utilized the two-dimensional aspect of this dynamical system to map to two different musical parameters, frequently pitch and rhythm. This study employs a different strategy. Instead of treating it as a two-dimensional map, the system can be rewritten in a single dimension as a recurrence relation:

$$x_{(n+1)} = 1 - a(x_n)^2 + bx_{(n-1)} \quad (3)$$

In other words, a single variable with two time delays.

2. INSTRUMENTATION

Au Revoir, Svetozar is scored for string quartet (two violins, viola, and cello) and guzheng. The piece was composed in 2016.

Composers typically call for players to tune the guzheng, a traditional Chinese zither, to a pentatonic scale. In this case, the guzheng required is the standard 21-string instrument. Chang [9] indicates that the most common tuning of the guzheng is a pentatonic scale in D – D, E, F#, A, B. See Figure 1 below. However, the scordatura tuning required for *Au Revoir, Svetozar* lowers the third and fifth pitches by a half-step to produce a D, E, F, A, Bb pentatonic scale.

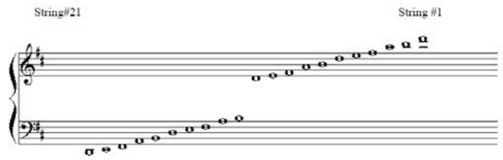


Figure 1. A typical tuning of the guzheng.

The piece does not require the guzheng player to use his or her left hand on the left side of the sliding bridges to alter a pitch or produce a pitch not available in the pentatonic tuning, as is common in much guzheng music. The player is free to employ both hands on the right-hand side of the bridges in order to facilitate performance of the leaps and the harp-like, discrete-pitch glissandi required in the piece. Throughout the piece, the guzheng strings should be allowed to vibrate freely after the player plucks the strings.

The piece requires the violin, viola, and cello players to use the standard tunings and produce sounds through standard bowings and pizzicato techniques.

Au Revoir, Svetozar often features three layers or types of sound, with specific instruments assigned to each timbral grouping. The guzheng produces one layer of the texture using the plucked sounds of individual strings and discrete-pitch glissandi. The viola produces another layer of the texture using standard bowing. The two violins and cello together generate a third layer often employing a pizzicato method of sound production and, at other times, a traditional bowing.

3. INITIAL CONDITIONS

The initial conditions selected for *Au Revoir, Svetozar* varied by instrumental part. For the two violins and cello, the initial conditions were

$$x_0=1.1 \text{ and } y_0 =1.1,$$

and the parameters were

$$a = 1 \text{ and } b = 0.3.$$

For the viola part, the initial conditions were

$$x_0 = 1.2 \text{ and } y_0 = 1.1,$$

and the parameters were

$$a = 1 \text{ and } b = 0.3.$$

For the guzheng part, the initial conditions were

$$x_0 = 1 \text{ and } y_0 =1,$$

and the parameters were

$$a = 1.4 \text{ and } b = 0.3.$$

These conditions produced, in each case, non-stable orbits that generated a stream of non-repeating (chaotic) values.

4. RESULTING OUTPUT

The computer program manipulated the resulting output by multiplying by 10, rounding, and taking the absolute value of the product. This process was employed in order to generate a set of integers that could be mapped easily to the musical parameters. The resulting output consisted of 15 possible different integers. The integers ranged from 0 to 14. Over the course of the values needed and generated for this particular piece, only one consistent or large-scale repetition of material was observed. That repetition occurs at the very end of the piece in the guzheng part.

5. MAPPING

The mapping employed for these output values varied depending on the instrumental group as described above under "Instrumentation". The mapping treated the x and y values individually as one-dimensional output. So, rather than mapping the x value to pitch and the y value to duration (for example), both the x and y values were mapped to a single music parameter.

5.1 Mapping for the violins and cello

The mapping for the violins and cello matched the output to pitch class. This mapping freed the composer to choose the octave appropriate for the particular instrument and musical situation. Table 1 shows the integer value and corresponding pitch-class result. Note that because of the number of possible outcomes and the chosen mapping, the likelihood for certain pitch-class results were greater than for others.

Output Value	Pitch Class
0	C
1	C#
2	D
3	D#
4	E
5	F
6	F#
7	G
8	G#
9	A
10	A#
11	B
12	C
13	D
14	E

Table 1. Mapping for violins and cello parts.

In this case, the likelihood of pitch classes C, D, and E appearing in the parts for violins and cello are increased.

5.2 Mapping for the viola

For most of the piece, *Au Revoir, Svetozar,* the viola is assigned the task of presenting a cantus firmus. In this instance, the cantus firmus employs the virelai, *Douce Dame*, by Guillaume de Machaut (c.1300-1377). This cantus firmus part utilizes the pitches from the original melody, but the mapping of the Henon Attractor determines the durations for each pitch.

Table 2 shows the durational values assigned to value generated by the Henon Attractor.

The piece is notated in 7/2 and 7/8 and the durations chosen reflect both those meters and the frequently long

durations of individual pitches in medieval canti firmi. The term "wild card" indicates a choice left completely open to the composer. The one "wild card" mapping in the case of *Au Revoir, Svetozar* allows the composer some flexibility in solving any musical problems encountered in the structure of the piece. The relatively rare instance of a "0" output also provides some flexibility in solving any notational problems that might arise in the score or parts.

Output Value	Duration Value
0	"Wild Card"
1	Quarter
2	Half
3	Dotted Half
4	Whole
5	Whole + Quarter
6	Dotted Whole
7	Dotted Whole + Quarter
8	Double Whole
9	Double Whole + Quarter
10	Double Whole + Half
11	Double Whole + Dotted Half
12	Dotted Double Whole
13	Dotted Double Whole + Quarter
14	Dotted Double Whole + Half

Table 2. Mapping for viola rhythms.

5.3 Durational mapping for the violins

In the second half of the piece, one section occurs in which the viola part shifts to a pizzicato pitch mapping of the Henon Attractor output. While this happens, the two violin parts take over the cantus firmus pitch material, but move with independent durations for each pitch as determined by output of the Henon Attractor.

5.4 Mapping for the guzheng

The mapping for the guzheng matched the integer output of the computer program not to pitch classes, but to specific pitches. The mapping is shown below in Figure 2.

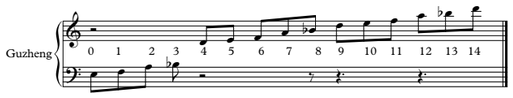


Figure 2. Mapping of the output to specific pitches for the guzheng.

This particular mapping does not, therefore, encompass the entire range of the guzheng.

6. COMPOSITIONAL RESULTS

Au Revoir, Svetozar is a single-movement work that has an overall duration of approximately ten minutes. The main sections of the piece follow a very brief introduction that features a pitch-discrete glissando gesture on the guzheng.

The first main section has as its unifying structure the cantus firmus in the viola part, which employs the pitch material from the Machaut melody. This first main section consists of six subsections that alternate between pizzicato parts in the violins and cello, rests in the violin and cello parts, and traditional bowing in the violin and cello parts.

The second main section begins after the completion of the cantus firmus melody in the viola part. This section, as mentioned before, features the two violin parts each presenting a different version of the cantus firmus using independent durations for each pitch in the Machaut melody as determined by the mapping of the Henon Attractor. The second main section consists of three subsections, with the second of these subsections containing pizzicati in all of the string quartet parts. The third and final subsection features pizzicati in all of the string quartet parts except the first violin part, which requires traditional bowing technique to present part of the cantus firmus.

Table 3 shows a possible listing of the piece's form.

Introduction		guzheng
Main Section 1		
	Subsection A	CF, pizz
	Subsection B	viola, guzheng
	Subsection C	vls, cello, bow
	Subsection D	pizz
	Subsection E	bow
	Subsection F	pizz
Main Section 2		
	Subsection G	violins, CF
	Subsection H	pizz
	Subsection I	vl1 bow CF, pizz

Table 3. Form of *Au Revoir, Svetozar*.

The abbreviations are: CF for cantus firmus, vls for violins 1 and 2, pizz for pizzicato, vl1 for violin 1, and bow for bowed parts.

6.1 Algorithmic Results

The use of the Henon Attractor as single dimension source of data produced excellent results. The values provided by this algorithmic technique were always useful for the individual instrument(s) assigned to present the program output. This particular type of mapping of the dynamical system's output sometimes created significant leaps between values in the musical result. That aspect of the output merely served as a feature of the dynamical system.

6.2 Aesthetic Results

Aesthetically, this multi-modal use of the Henon Attractor works well. Utilizing the output in different ways produces strong and interesting contrasts. Particularly effective is the highlighting of the cantus firmus melody against the pizzicato passages in the violins and violoncello. The juxtaposition of the cantus firmus melody in either the viola or the violins using traditional bowing technique with the pentatonic leaps in the guzheng also serves to spotlight each part at different times in the course of the piece.

Selected for a performance at the EchoFluxx Festival in Prague, the Hong Kong New Music Ensemble will premiere *Au Revoir, Svetozar* at the Festival in the month of May 2016.

Figure 3 shows twenty-six measures from the piece. Those measures exhibit various aspects of the textures, pitches, and rhythms generated by the algorithm.

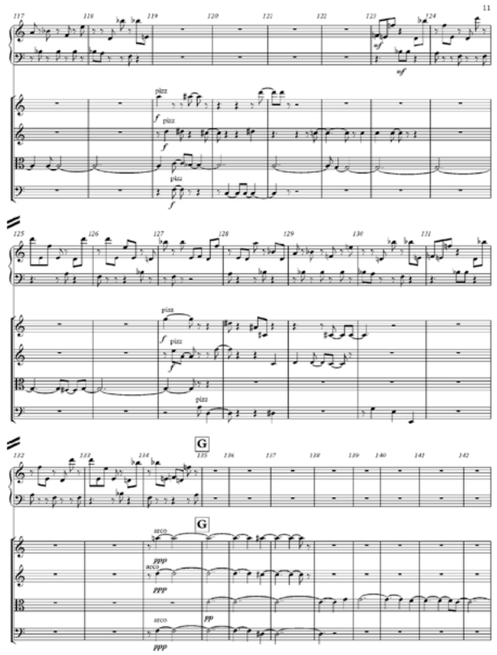


Figure 3. Measures 117-142 of the score for *Au Revoir, Svetozar*.

Five measures in Figure 3 (bars 119-123) show the continuing cantus firmus in the viola part while the violins and violoncello complete a pizzicato passage. Measures 117-118 and 124-126 show the juxtaposition of the guzheng pitch material generated by the algorithm with the ongoing cantus firmus in the viola and without the pizzicati in the violins and cello. One measure in the example (bar 127) shows all three textures occurring simultaneously, while measures 132 and 133 show a brief transition that features only the guzheng and leads to a new texture in which all of the string quartet instruments bow a passage.

7. CONCLUSION

The two-dimensional dynamical system known as the Henon Map can be effectively mapped to a single musical parameter in order to produce convincing new art-music compositions. The output can be mapped to pitch or rhythm successfully. A simultaneous multi-modal use of the Henon Attractor appears particularly compelling.

Future work in this area could include mapping the system's output to other musical parameters such as articulation or timbre and continued experiments with a multi-modal approach – simultaneously controlling more than one musical parameter.

8. REFERENCES

[1] M. Henon, "A two-dimensional mapping with a strange attractor," *Communications in Mathematical Physics*, vol. 50, no. 1: pp 69-77, 1976.

[2] M.J. Islam, M.S. Islam, and M.A. Rahman, "Two Dimensional Henon Map with the Parameter Values 1<a<2, |b|<1 in Dynamical Systems," *Annals of Pure and Applied Mathematics*, Vol. 2, No. 2, pp.164-176, 2012.

[3] J. Pressing, "Nonlinear maps as generators of musical design," *Computer Music Journal*, Vol.12, No. 2: pp. 35-46, 1988.

[4] R. Waschka, "Computer-Assisted Composition and Performance: The Creation of *A Noite, Porém, Rangeu E Quebrou,*" *Leonardo Music Journal*, Vol. 2, No. 1: pp. 41-44, 1992.

[5] "Henon Map Melody Generator" http://henon.sapp.org, n.d.

[6] James Clark, *Advanced Programming Techniques for Modular Synthesizers*. Web book, 2003.

[7] A. E. Coca, D. C. Correa and Liang Zhao, "Computer-aided music composition with LSTM neural network and chaotic inspiration," *The 2013 International Joint Conference on Neural Networks*, Dallas, TX, 2013, pp. 1-7.

[8] R. Waschka and A. Kurepa, "Dynamical Systems in Two Music Compositions" in N. Mastorakis, editor, *Mathematics and Computers in Modern Science: Acoustics and Music,...* World Scientific Engineering Society, pp. 90-92, 2000.

[9] C. Chang, *Sound of China: Guzheng Basic Tutorial*. Sound of China Publishing, 2011.

From live to interactive electronics. *Symbiosis*: a study on sonic human-computer synergy.

Artemi - Maria Gioti
University of Music and Performing Arts Graz, Austria
Institute of Electronic Music and Acoustics
gioti@iem.at

ABSTRACT

Human-computer interaction in live electronics is – in most cases still today – a reflection of the instrumental model. The electronics, like a musical instrument, must react to the performer's orders as precisely as possible. The frequent use of control devices, such as musical interfaces, reinforces the instrumental character of live electronics, leading to an action-reaction performance model, derived from instrumental performance (performative electronics). A detachment from the instrumental model in live electronic music can be achieved through the design of human-computer interaction not as a reflective, but as a formative condition. Instead of taking the action-reaction model for granted, the relationship between the performer and the electronics can be re-examined and redefined. A design of human-computer interaction from a compositional point of view enables the transformation of the musical work into a sonic process, as the result of a reciprocal sonic interaction between man and machine (interactive electronics). This paper attempts a systematization of live electronics on the basis of (human or software) agency and describes the author's personal approach to the design of interactive sonic systems in a piece for double bass and interactive electronics.

1. FROM LIVE TO INTERACTIVE ELECTRONICS

1.1 Agency in Live Electronics

Although live manipulation of sound by electronic means has been an established practice for more than fifty years, the term "live electronics" is one of the most ambiguous terms in today's western art music. The practice of live electronic sound manipulation includes a vast number of possibilities concerning techniques, sound material and (human or software) agency, all summarized under the term "live electronics".

The ambiguity of the term becomes evident when it is placed in an electro-instrumental context: e.g.: "piano and live electronics". While an acoustic instrument, such as the piano, is clearly defined through its intrinsic (e.g. sound production, spectral characteristics) and extrinsic properties (e.g. historical repertoire), "live electronics" is a vague, technical indication that contains no information about the type of electronic manipulations applied or the resulting sound output.

Regarding sound material in works for acoustic instruments and live electronics, this can vary from pre-produced sounds (e.g. samples, pre-recorded and processed or synthesized material) to sounds generated in real-time and from computer-generated, purely electronic sounds to the transformation of an input signal in real-time. Techniques employed in live electronics include playback of pre-produced or live-recorded material, sound synthesis, signal processing and several audio effects, to name but a few.

However, while the material and techniques used in live electronics do not differ significantly from those employed in fixed media pieces, what is genre-specific to live electronics is agency. "Liveness" suggests the presence – *physical* or *psychological* [1] – of an agent, adjusting some kind of run-time control data [2]. The agent can be either human (one or more performers) or virtual (software agent) and its role can vary from a triggering function to a *reciprocal interaction* [3] within a dynamical sonic system.

Franklin and Graesser [4] define an agent as follows:

> *An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.*

According to Wooldridge and Jennings [5], the agent is a hardware or software-based computer system characterized by autonomy, social ability, reactivity and proactiveness ("the ability to exhibit goal-directed behavior by taking the initiative").

In philosophy, the term is generally used to describe an entity which is able to act intentionally within a certain environment [6]. This definition is the most applicable in live electronic music, since it encompasses all types of agency. Intention-driven action, whether initiated by a human or a software agent, is a prerequisite for "liveness" in electronic music. The type (human or virtual) and role of the agent in live electronic systems could therefore be regarded as the main criterion for further differentiation within the field of live electronics.

1.2 Human-Computer Interaction

Agency, as the key aspect of live electronic music, goes hand in hand with the concept of *interaction*, a term as vague and ambiguous as "live electronics".

For the moment, let's restrict our discussion of *interaction* to human-computer interaction. Human-computer interaction as an interdisciplinary research field among computer science, psychology, media theory and other disciplines focuses on the development of interfaces for the improvement of human-computer communication.

However, the term human-computer interaction is – from a philosophical point of view – a paradox. Interaction, as a series of actions connected through causal relationships, is only possible between two parts that are able not only to react, but also to act. A prerequisite for action is intention: there can be no action without intention and therefore, since machines have no intention, they cannot act. A machine can only react to the user's orders. It is operated by the user, but does not interact with him/her. Interaction in its literal sense is only possible among humans.

> *The miracle of human-computer interaction is that it is impossible as interaction in the true sense of the word. [...] The miracle is that humans were bold and intelligent enough to establish this. The miracle is not that machines were so intelligent to do it.* [3]

1.2.1 Human Agency: Performative Electronics

Any type of agency in live electronic systems requires some kind of human-machine communication.

In the case of a human agent, the performer (either the instrumentalist or a second performer, or sometimes even the composer) operates some software by setting control variables. The role of the human agent can vary from triggering presets to providing data streams in real-time. For this task the performer can either use general purpose interfaces (mouse, keyboard etc.) or specialized musical interfaces (MIDI-pedals, sensor-based interfaces etc.).

Miranda and Wanderley [7] classify musical interfaces as *augmented musical instruments*, *instrument-like gestural controllers*, *instrument-inspired gestural controllers* and *alternate gestural controllers*. The instrument analogy that is prominent in Miranda and Wanderley's classification is indicative of an interface design oriented towards musical instruments and the rather common opinion that the learning process of a musical interface should be similar to learning an instrument [8] [9]. This analogy is not only the by-product of a well-established music tradition [10], but also the result of a task oriented interface design, developed for problem solving in custom human-computer interaction. The human-computer interface is "an emergent event in the development of computers" [11], dictated by the need for information exchange between the user and the machine. The musical interface, like any other interface, is also a control device, used to operate a machine (computer) while performing a specific task (playing music).

Despite all efforts for an instrument-oriented design of musical interfaces (bio-sensor based interfaces being perhaps the only exception), the latter have been criticized for lack of musicality in communicating virtuosity and emotion [8]. Furthermore, the "instrument" metaphor is often considered as a rather debatable approach when used in the context of interactive music systems [2].

An instrument (whether musical or not) is designed to be operated, to be controlled. The instrument has no intention: it is designed to react and not to act. Consequently, the instrumental condition excludes any kind of interactivity. The control device is only used to "translate" the user's orders, so that they can be executed by the computer.

Human agency in live electronics is a synonym of the instrumental condition: the electronics react to the agent's orders according to the instrumental model, they are *performed* by the agent (*performative electronics*). In the communication between the agent (performer) and the software the distinction between subject and object is clear. The subject (agent) acts and the object (software) reacts. The subject-object communication is restricted to a one-way reaction, excluding human-computer reciprocity.

1.2.2 Software Agency: Reactive and Interactive Electronics

A reciprocal interaction between human and computer in live electronic systems is only possible in the context of software agency.

In this type of agency, the run-time control data required for the live electronics is provided by the software itself. This requires decisions made by the computer either randomly or based on the analysis of current input data, or in most cases both.

In a piece for acoustic instruments and live electronics the input data can be provided by the instrumental sound. The signal of the acoustic instrument(s) undergoes various analyses (FFT, Amplitude Tracking, Onset detection etc.) in real-time and is used as a source for control data for the electronics algorithm. The algorithm then adjusts its output to the current input, constantly adapting to changing external conditions. Electronics of this type are *reactive*.

In *reactive electronics*, the subject-object communication (performer-software) is replaced by an object-object interaction (sound-software). The system is reactive but not interactive. Although the live electronics react and adapt to the input data provided by the instrumentalist, the performer does not adapt his/her reactions to the output of the electronics.

Interactive electronics require mutual adaptation among system components (in this case the performer and the algorithm). Such a reciprocal interaction between the performer and the electronics can only be achieved through the use of some kind of algorithmic score, allowing the performer to react to the electronics in the same way that the electronics react to the performer.

In interactive electronics, both system components analyze each other's actions and react correspondingly. The object-object interaction is transformed into a subject-subject interaction. The computer is able to analyze human actions, make decisions and act upon intention. In other words, the software is turned into a "subject" able to interact with the human user in an equal and reciprocal interaction.

According to Nake [3], the interface is the coupling of surface and subface, surface being the intentional interpretant (e.g. computer screen) and subface the causal interpretant (e.g. display buffer). According to this definition, the communication between the two system components in interactive electronics is shifted from the surface to the subface. The interface – or any other control unit – is replaced by a direct and unmediated interaction, based exclusively on sound. The relationship between the performer and the computer is not a mere translation of the performer's intention, but a causal chain of reciprocating actions between the performer and the computer: the mediation is not *intentional*, but *causal*.

1.3 Systemic Composition: Interaction as a Compositional Approach

The transition from intentional to causal mediation in live electronics can be the starting point for an expansion of the scope of the compositional process in general.

A human-computer interaction that is based on intentional interpretation limits live electronics to an instrumental behavior. The computer simply "translates" the user's intention into a sound output. In this *action-sound relationship* [10] there is a clear distinction between intention and action on the one hand and interpretation and reaction on the other hand. The first two belong to the human agent and the second two to the computer. This approach restricts live electronics to a performative character, requiring a human agent that operates the system, but does not interact with it (performative electronics).

In a human-computer interaction based on causal mediation on the other hand, the *action-sound* relationship is transformed into an *action-action* relationship, through the replacement of *intention* with *causality*. The computer does not simply "translate" the user's intention into a sound output. The actions of both the performer and the computer are in a causal and reciprocal relationship with each other, forming a network of interrelations.

This transition from control to interactivity enables the creation of self-organizing sonic systems, the sound output of which is the result of mutual dependencies between the performer and the electronics. The concept and practice of interactive electronics inevitably leads to a redefinition of the compositional process. The composer's task is shifted from composing sounds to composing sonic interactions. The creation of a network of sonic interrelations between the musician and the electronics becomes part of the compositional process, resulting in a new understanding of composition as a practice and of the musical work itself.

> *By delegating some of the creative responsibility to the performers and a computer program, the composer pushes composition up (to a meta-level captured in the processes executed by the computer) and out (to the human performers improvising within the logic of the work).* [12]

At this point, an important distinction needs to be made. A music system cannot be called interactive, unless every action performed within it is an intentional response to the previous one. This is the main difference between interactive and reactive systems, in which only every second action is an intentional response to the previous one.

Having said that, most definitions of interactive music systems seem to be problematic. According to Rowe, "interactive music systems are those whose behavior changes in response to musical input. Such responsiveness allows these systems to participate in live performances, of both notated and improvised music" [13]. Rowe's definition clearly refers to reactive systems, confusing responsiveness with reciprocity.

> *Rowe presents an image of a computer music system listening to, and in turn responding to, a performer. The emphasis in Rowe's definition is on the response of the system; the effect the system has on the human performer is secondary.* [14]

The same can be argued for Chadabe's definition of interactive composing. Chadabe regards interaction as an improvisational and not as a compositional approach. His model is composed of a performance interpretation algorithm and a response algorithm (consisting of a composition and a sound algorithm), both regulating the response of the computer to the performer [15]. In Chadabe's model there is no algorithm determining the performer's response.

Similar efforts have been made in the field of Artificial Intelligence, aiming at stylistic imitation through machine learning. Projects like PAPAGEI (by S. Bakht and C. Barlow) [16], OMax (developed at IRCAM among others by G. Assayag) [17] and Voyager (by G. E. Lewis) [18] generate improvisational material, based on real-time machine learning. Of course, this one-sided imitation is far from being a reciprocal interaction. The computer simply imitates the musician, by generating variations of the material played by him/her. The system may have intelligence, but it has no intention. Moreover, the pitch-based approach that is followed in these approaches is a huge limitation of the possibilities of electronic sound generation.

An interesting compositional approach in a neighboring – if not overlapping - field to interactive electronics are Di Scipio's audible ecosystems. Nevertheless, in audible ecosystems the focus of interaction lies in the triangular connection between the human agent, a DSP unit and the sonic ambience [2] and not in human-computer reciprocity. Another important difference between interactive electronics and audible ecosystems is that, in the latter, the object of human-machine communication is not necessarily sound. The human agent can control the electronics via control devices, resulting in a mediated (interface-based) and not sound-based interaction, a condition that has been described in this paper as *performative electronics*.

Interaction as a compositional approach is generally a rather new approach to live electronics that remains to be explored further, not only compositionally, but also in a conceptual and theoretical basis. A reassignment of roles and a redefinition of the relationship between man and machine in live electronic systems can be the starting point for a new understanding of composition, that regards technology as a formative and not as a reflective condition.

In the following chapter I describe personal work that illustrates a compositional approach to sonic human-computer interaction.

2. *SYMBIOSIS* (2015), FOR DOUBLE BASS AND INTERACTIVE ELECTRONICS

Symbiosis, for double bass and interactive electronics, is a study on live electronics as sonic human-computer interaction. The piece is an autonomous sonic system, the output of which is determined by the reciprocal interaction between the double bass performer and the computer.

As a composed system, *Symbiosis* cannot be regarded – and thus notated – as a linear sequence of sound events. The score of the piece consists partly of conventional notation and partly of abstract and improvisational notation, while the electronics run independently during the performance, based on self-regulating processes and do not require a second performer.

In *Symbiosis*, the changing interrelations between the performer and the software agent are the focus of the compositional process and the piece itself is considered as a sonic process. Instead of temporal sonic structures, the work is composed of a non-temporal, abstract interaction model, the output of which is the result of sonic human-computer synergy. This model undergoes several modifications during the piece, leading to various stages of interaction and reciprocating control. The interaction model of the piece is explained in detail below.

2.1 Interaction Model

The interaction model of *Symbiosis* consists of two discrete algorithms, which are responsible for the action-reaction cycle between the acoustic instrument (real sound object) and the electronics (virtual sound object). The electronics algorithm adjusts the output of the electronics to the live input, while the score algorithm is responsible for the performer's reaction to the electronics, providing abstract models as possible continuations of the score. The two algorithms correspond to different structural levels, the electronics algorithm determining the parameters of single sound events (microstructure) and the score algorithm being responsible for larger sections of the piece (macrostructure). Combined, the electronics and score algorithm enable a reciprocal communication between the performer and the computer, based on an action-analysis-reaction loop.

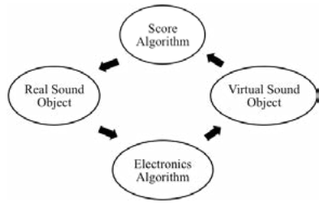


Figure 1. The Interaction Model of *Symbiosis*.

2.1.1 The Electronics Algorithm

The electronics algorithm is an adaptive algorithm, consisting of an analysis, a mapping and a sound generation stage. In the first stage, the input signal of the double bass is being analyzed. Data derived from FFT analysis and amplitude tracking is then being mapped, in order to provide the parameters for the virtual (electronic) sound. The algorithm constantly adapts to current input, changing its output correspondingly. The choice of a mapping strategy is therefore decisive for the behavior of the system, as it determines the interconnections between its components and, consequently, its output.

Another characteristic of the electronics algorithm is its non-linearity. The complex dependencies between the components create a system which is highly sensitive to initial conditions and, thus, non-deterministic.

2.1.2 Score Algorithm

Based on the output of the electronics, the performer follows a set of rules on how to proceed with the score reading (algorithmic score). The score provides the performer with an open form, the interpretation of which is based partly on the non-deterministic output of the electronics and partly on decisions made by the performer in real-time. The algorithmic score model was used in the last part of the piece and marks the transition from reaction to interaction. As the behavior of the virtual sound object becomes more complex and less predictable, the performer's role is shifted from action to interaction, allowing reciprocity and mutual adaptation.

Modifications of variables (e.g. input signal) or processes (e.g. mapping) within the above-mentioned interaction model affect the behavior of the system, resulting in different control levels and interaction stages. Such variations were applied in a macro-structural level and are responsible for the formal development of the piece.

Symbiosis can be divided in three parts, each illustrating a different stage of interaction between the two system components (performer-computer). The transition from reaction to interaction is thematized as a formal process and is reflected not only in the changing interdependencies of the system, but also in its sonic properties. The strictly reactive and relatively predictable behavior of

the electronics in the first part of the piece is replaced by a much more complex and less transparent mapping process in the second part, until the virtual sound object is transformed into an autonomous entity (third part).

2.2 Sound Material

The electronics part of *Symbiosis* is based on real-time Convolution of the input signal (double bass) with artificial, pre-composed Impulse Responses (IRs). The Impulse Responses used in the piece are transformations of a real Room Impulse Response (RIR), measured in György-Ligeti-Saal of the Haus für Musik und Musiktheater (MUMUTH) in Graz [19]. The Impulse Response of the concert hall was only used as a reference. The resulting IRs were derived from abstract processes and do not correspond to the spatial properties of the concert hall or any other existing physical space. The composed Impulse Responses were based on the superposition of delayed and reverted copies of the initial IR and have a rather gestural, than spatial character. The delay time and number of superimposed copies are gradually increased during the piece, leading to an escalation of the temporal and spectral autonomy of the virtual sound object.

2.3 Formal Development

2.3.1 Adaptive Signal Processing

In the first part of the piece, data derived from FFT analysis of the input signal is used in order to control an adaptive signal processing algorithm.

First, the input signal is analyzed in order to extract the spectral centroid. The value of the centroid is then mapped to an amplitude value, determining the amount of signal that is sent to the convolution. If the centroid value is lower than a certain threshold, then no signal is sent to the signal processing algorithm. This means that the input signal is processed selectively and dynamically, according to its spectral characteristics.

The spectral centroid is a measure for the energy distribution within a spectrum. Therefore, not only pitch, but also playing techniques can affect its value, by determining how “pure” or “noisy” a sound is. The value of the spectral centroid is highly unpredictable, partly because the energy in the higher frequencies is more than in the lower frequencies (meaning that the choice of a less or more “noisy” playing technique can be decisive) and partly because of its dependency on the type of microphone, its distance from the instrument and several other factors.

The sound material of the instrumental part points to the non-deterministic character of the analysis stage. The material for the double bass was chosen based on spectral content. Contrasting sound spectra and playing techniques that enable transient overtone accentuation were used in order to ensure variability. The score in this part of the piece consists of long durations and playing techniques that enable subtle spectral variations, subject to non-deterministic dynamics.

The panning of the output signal to the four output channels is controlled by its amplitude.

In this part of the piece, the sonic system consists of an acting (performer) and a reacting component (electronics), both following deterministic instructions (score and mapping process respectively) and yet resulting in a non-deterministic sonic synergy.

2.3.2 Increasing System Complexity

In the second part, the complexity of the system is increased through less transparent mapping processes and a non-deterministic system behavior. More than one (from three to six) Impulse Responses are used simultaneously and the algorithm cross-fades among them based on spectral information.

Another characteristic of the electronics algorithm in the second part of the piece is the introduction of feedback as a self-regulating structural process. This feedback is not electroacoustic (between the loudspeakers and the microphone), but is implemented in the signal processing algorithm. The output of each convolution is fed back to all the others and to itself, forming a complex feedback matrix. The feedback matrix is the first step in the transformation of the virtual sound object into an autonomous sound organism, providing additional input to that of the acoustic instrument and, thus, increasing system autonomy.

Another aspect of feedback that plays a decisive role in the detachment of the virtual sound object from the real is its non-deterministic character. A self-organizing and self-controlling process, such as a feedback loop, is extremely sensitive to initial conditions and cannot be controlled externally.

Interaction between the real sound object and the feedback matrix of the electronics lies in a rather non-transparent mapping between the spectral centroid and amplitude values within the matrix. However, the values of the spectral centroid are not mapped directly to amplitude values. A cumulative sum of the centroid values is calculated by adding each new value to the stored one, after weighting them with different factors. By giving greater weight to past information the weighted sum value changes slower than the actual centroid values, resulting in slow amplitude fluctuations within the matrix. The performer’s control over the electronics algorithm has become mediated, indirect and deferred in time.

While the mapping process increases in complexity, enabling indeterminacy, the instructions in the score remain deterministic. The musician is asked to play a thoroughly notated, pointillistic texture, while the electronics are becoming less and less responsive and are gradually transformed into an autonomous system. The interaction paradox of the second part points to the necessity of a reciprocating communication between the performer and the software agent, which takes place in the third part of the piece.

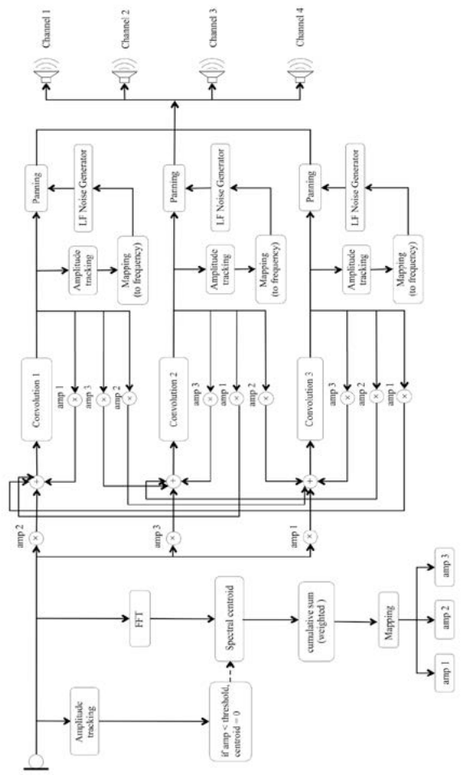


Figure 2. Signal flow chart (part 2)

2.3.3 Human-Computer Reciprocity

In the third part of *Symbiosis*, the signal processing chain is interrupted and for the first time the input material is provided by the computer and not the double bass. Through the replacement of the input signal with an impulse generator, the computer is turned into an acting component, able not only to process, but also to generate sound. The impulses are controlled by a noise generator and fed into the convolution and feedback matrix. The amplitude values for the matrix are no longer controlled externally (by the double bass signal), but internally. The algorithm sets an amplitude value for the feedback matrix, which in this way evolves into an autonomous, self-regulating system. As in the first and second part, the panning of the output signal is controlled by its amplitude.

Since both input and control data required for the electronics algorithm are provided by the computer, every direct dependency on the instrumental sound is cut off. The third part of *Symbiosis* is the last stage in the development of a reactive into a self-organizing system.

The changed status of the virtual sound object is facilitated by a new interaction model and a redefinition of the roles of the two interacting components (performer-electronics). Instead of providing run-time control data for the electronics algorithm, the performer can only intervene in the system and force it to respond. More specifically, if the weighted cumulative sum of the spectral centroid is higher than a threshold value, then the output of the electronics is set to zero. When the amplitude of the electronics recovers its initial value, the same process is repeated, this time with different weighting factors, resulting in a stronger smoothing of the analysis data.

The score in this part of the piece is based on an abstract notation that allows the performer to adapt to the electronics, by adjusting his/her reactions in real-time. No pitch or duration is notated, while the form is open, meaning that the notated actions can be performed in any order, one or more times. The performer tries to eliminate the continuously accumulating feedback by playing material that will raise the centroid sum. Every time that he/she succeeds, the weighting factors are reset making the next effort more difficult. When the performer’s efforts fail to meet the criteria established by the algorithm, the piece is terminated.

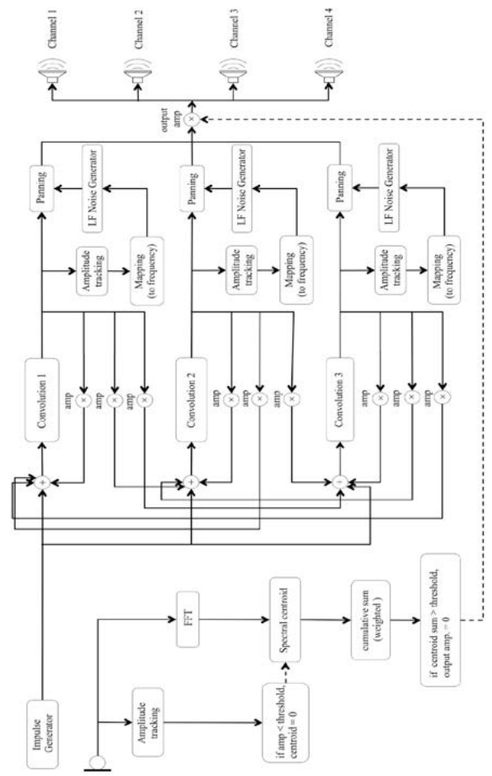


Figure 3. Signal flow chart (part 3).

2.4 Conclusion

Symbiosis (from Greek σύν, “with”, “together” and βίος, “life”) illustrates a shift in control from man to machine. The performer (subject) gradually loses control over the computer (object), until human-computer communication becomes unresponsive. Starting from a direct control over

the electronics which gradually becomes mediated, the performer's role is finally restricted to a reset function, until he/she loses every influence upon the system. The electronics, on the other hand, evolve from a mere reflection of the instrumental sound into an independent, self-organizing sonic system. This shift in control is reflected not only in the interaction model and the transition from reaction to reciprocal interaction, but also in the input and control data of the system. The live input of the double bass and the control data derived from FFT analysis and mapping is gradually replaced by computer generated impulses and decisions made by the software agent.

The mapping process is also modified, in order to facilitate the transition among different control levels and interaction stages. The use of the cumulative sum as a data smoothing process is an example of a mapping strategy that enables causal instead of intentional interpretation. A direct mapping of input to control data is a simple translation of an input into an output. With the introduction of a data smoothing process, however, the input data – and together with it the user's intention – is filtered. The output of the system is in a causal relation to the user's actions, but it is not a direct translation of his/her intentions. The user's intention is filtered through the intention of the system: the computer interprets input data according to its own "will".

A software agent that acts upon intention is a prerequisite for reciprocal human-computer interaction. Interaction can only take place among interacting parts of the same status (subject-subject). Therefore, a passive reaction of the system to the user's orders establishes it as an object and restricts human-machine communication to a control relationship. Intentional agency, on the other hand, transforms the object (computer) into a subject and brings human-computer communication to the stage of reciprocal interaction. In this stage, both system components are capable of analyzing each other's actions, making decisions and expressing intention through action. The distinction between subject and object is lifted.

3. REFERENCES

[1] S. Emmerson, *Living electronic music*, Aldershot, Hampshire: Ashgate, 2007.

[2] A. Di Scipio, "Sound is the interface: From interactive to ecosystemic signal processing", *Organised Sound,* vol. 8, no. 3, pp. 269-277, 2003.

[3] F. Nake, "Surface, Interface, Subface: Three Cases of Interaction and One Concept", in *Paradoxes of Interactivity: Perspectives for Media Theory, Human-Computer Interaction and Artistic Investigations*, U. Seifert, J. H. Kim and A. Moore, Eds., Bielefeld: Transcript Verlag, 2008, pp. 92-109.

[4] S. Franklin and A. Graesser, "Is it an agent, or just a program?: A taxonomy of autonomous agents", in *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996.

[5] M. Wooldridge and N. Jennings, "Agent Theories, Architectures, and Languages: a Survey", in *Intelligent Agents,* M. Wooldridge and N. Jennings, Eds., Berlin: Springer-Verlag, 1995, pp. 1-22.

[6] D. Davidson, *Essays on Actions and Events,* Oxford: Oxford University Press, 2001.

[7] E. R. Miranda and M. Wanderley, *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*, Middleton: A-R Editions, 2006.

[8] J. McDermott et al., "Should Music Interaction Be Easy?", in *Music and Human-Computer Interaction,* London: Springer, 2013, pp. 29-47.

[9] J. Ryan, "Some remarks on musical instrument design at STEIM", *Contemporary Music Review*, vol. 6, no. 1, pp. 3-17, 1991.

[10] W. Brent, "Perceived Control and Mimesis in Digital Musical Instrument Performance". Available:http://cec.sonus.ca/econtact/14_2/brent_mimesis.html (accessed: 13.02.2016).

[11] S. Card, T. Moran and A. Newell, *The Psychology of Human-Computer Interaction*. Boca Raton: CRC Press, 1983.

[12] R. Rowe, "The Aesthetics of Interactive Music Systems", *Contemporary Music Review*, vol. 18, no. 3, pp. 83–87, 1999.

[13] R. Rowe, *Interactive Music Systems: Machine Listening and Composing*, London: MIT Press, 1993.

[14] J. Drummond, "Understanding Interactive Systems", *Organised Sound,* vol. 14, no. 2, pp. 124-133, 2009.

[15] J. Chadabe, "Interactive Composing: An Overview", *Computer Music Journal*, vol. 8, no. 1, pp. 22–27, 1984.

[16] S. Bakht and C. Barlow, "PAPAGEI: An Extensible Automatic Accompaniment System for Live Instrumental Improvisation", in *Proceedings of the 2005 International Computer Music Conference*, Montreal, 2009, pp. 521-523.

[17] S. Bubnov, G. Assayag, O. Lartillot and G. Bejerano, "Using Machine-Learning Methods for Musical Style Modeling", *IEEE Computer*, vol. 10, no. 38, 2003.

[18] G.E. Lewis, "Too Many Notes: Computers, Complexity and Culture in "Voyager"", *Leonardo Music Journal,* vol. 10, pp. 33-39, 2000.

[19] G. Eckel and M. Rumori, "StiffNeck: The Electroacoustic Music Performance Venue in a Box," in *Proceedings of the 40th International Computer Music Conference and 11th Sound and Music Computing Conference*, Athens, 2014, pp. 542-546.

A Web-based System for Designing Interactive Virtual Soundscapes

Anıl Çamcı, Paul Murray and Angus G. Forbes
University of Illinois at Chicago
Electronic Visualization Lab
[acamci, pmurra5, aforbes]@uic.edu

ABSTRACT

With the advent of new hardware and software technologies, virtual reality has gained a significant momentum recently. VR design tools, such as game engines, have become much more accessible and are being used in a variety of applications ranging from physical rehabilitation to immersive art. These tools, however, offer a limited set of tools for audio processing in 3D virtual environments. Furthermore, they are platform-dependent due to performance requirements and feature separate editing and rendering modes, which can be limiting for sonic VR implementations. To address these, we introduce a novel web-based system that makes it possible to compose and control the binaural rendering of a dynamic open-space auditory scene. Developed within a framework of well-established theories on sound, our system enables a highly detailed bottom-up construction of interactive virtual soundscapes by offering tools to populate navigable sound fields at various scales (i.e. from sound cones to 3D sound objects to sound zones). Based on modern web technologies, such as WebGL and Web Audio, our system operates on both desktop computers and mobile devices. This enables our system to be used for a variety of mixed reality applications, including those where users can simultaneously manipulate and experience a virtual soundscape.

1. INTRODUCTION

Sound is an inherently immersive phenomenon. The air pressure originating from a sound source propagates in three dimensions. Although music is considered primarily a temporal art, the immersive quality of sound has been exploited throughout music history: in ancient antiphons, different parts of the music were sung by singers located at opposing parts of a church to amplify the effect of the call-and-response structure[1]. In the 1950s, the composer Karlheinz Stockhausen composed one of the first pieces of quadraphonic music using a speaker placed on a rotating table surrounded with 4 microphones. When played back, the resulting recording would envelope the listener with swirling gestures. Since the 1950s, many sound art pieces have highlighted the spatial qualities of sound by exploring the continuities between music and other art forms such as painting and sculpture.

In recent years, immersive media has been gaining popularity with the advent of new technologies such as commercial depth-tracking devices and head-mounted displays. Accordingly, software tools to create immersive media has become more accessible. Many artists today, for instance, use game engines to create virtual reality artworks. However, modern immersive design tools heavily favor the visual domain. Despite many studies that have highlighted the role of audio in improving the sense of immersion in virtual realities [2, 3], audio processing in modern game engines remain an afterthought. We have previously discussed a sound-first VR approach based on well-established theories on sound objects and soundscapes [4]. Building up on the taxonomy introduced in this study, the current paper introduces a novel web-based system that enables the rapid design of both virtual sonic environments and the assets (i.e., sound objects and sound zones) contained within them. Specifically, our system:

- provides a user-friendly 3D environment specific to sonic virtual realities, with specialized components such as sound objects and sound zones;
- offers both interactive and parametric manipulation of such components, enabling a precise control over highly-detailed virtual soundscapes;
- introduces a multi-cone model for creating 3D sound objects with complex propagation characteristics;
- enables adding dynamism to objects via hand-drawn motion trajectories that can be edited in 3D;
- makes it possible to manipulate virtual sonic environments at various scales using multiple view and attribute windows;
- offers a unified interface for the design and the simulation of such environments, allowing the user to modify a sound field in real-time;
- operates on the web-browser so that it supports mobile devices, which therefore makes it possible for the user to simultaneously explore and edit augmented sonic realities.

2. RELATED WORK

2.1 Sound in Virtual Reality

Modern VR design tools, such as game engines, offer basic audio assets, including point sources and reverberant zones. These objects are created and manipulated through

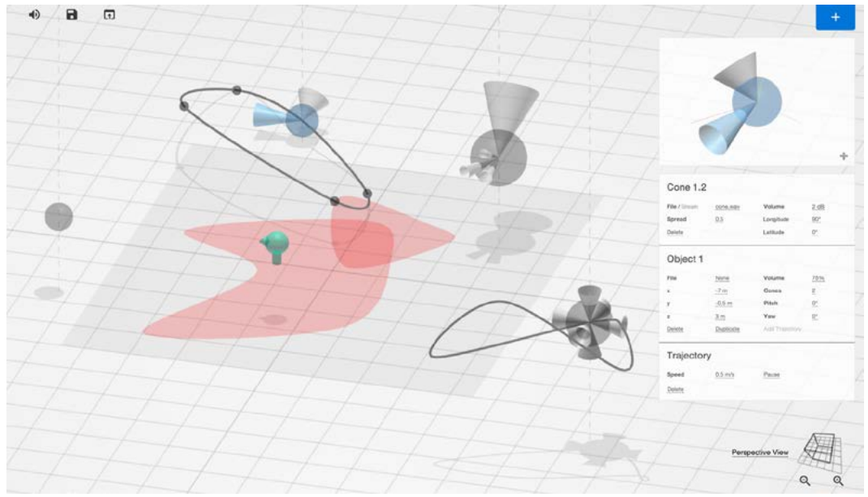


Figure 1. A screenshot of our user interface on a desktop computer displaying an object with two cones and a motion trajectory being edited. On the top right region, a close-up window displays the object with the cone that is currently being interacted with highlighted in blue. The windows below this close-up allows the user to control various attributes of the cone, the parent object, and its trajectory. Two overlapping sound zones are visualized with red polygons. A gray square represents the room overlay. The user is represented with a green dummy head.

the same interactions used for visual objects on these platforms. Additionally, third-party developers offer plug-ins, such as 3Dception[1], Phonon 3D[2] and RealSpace3D[3], that extend the audio capabilities of these engines with such features as occlusion, binaural audio, and Ambisonics. However, these extensions act within the UI framework of the parent engine, and force the designer to use object types originally meant to describe graphical objects, which can be limiting for sound artists.

Other companies specialize in combined hardware and software VR solutions. *WorldViz*, for instance, offers an "Ambisonic Auralizer" consisting of a 24-channel sound system, which can be controlled with Python scripts using their VR design platform called Vizard[4]. Although their tools have powerful spatializing capabilities, no user interfaces exist for creating sonic environments using them.

The *Zirkonium* software developed initially for the *Klangdom* surround sound system at the ZKM Institute for Music and Acoustics, allows the design of multiple spatial trajectories for sound sources [5]. Furthermore, the software allows the parametric and temporal manipulation of these trajectories.

IRCAM's *Spat* software[5] enables the creation of dynamic 3D scenes using binaural audio and Ambisonics. Although *Spat* provides a comprehensive set of tools which can be used to develop 3D audio applications within the Max programming environment, it does not offer a singular interface for virtual environment design.

SoundScape Renderer [6], developed by researches at the Quality and Usability lab at TU Berlin, is a system for the positioning of sound sources around a stationary listener using a 2D overview of the scene. Users of this software can assign arbitrary sound files and input sources to virtual objects. The SoundScape Renderer offers advanced rendering techniques, such as WFS, VBAP, Ambisonics as well as binaural audio.

[1] https://twobigears.com/3dception.php
[2] https://www.impulsonic.com/products/phonon3d/
[3] http://realspace3daudio.com
[4] http://www.worldviz.com/products/vizard
[5] http://forumnet.ircam.fr/product/spat-en

2.1.1 Web Audio API

The Web Audio API [7] is a JavaScript library for processing audio in web applications. A growing number of projects utilize this tool due to its high-level interface and its ability to operate on multiple platforms. Using the Web Audio API, Rossignol et al. [8] designed an acoustic scene simulator based on the sequencing and mixing of environmental sounds on a time-line. Lastly, Pike et al. [9] developed an immersive 3D audio web application using head-tracking and binaural audio. The system allows its users to spatialize the parts of a musical piece as point sources in 3D. These examples demonstrate that Web Audio is powerful enough to be used as a back end for sonic virtual realities.

Our implementation utilizes the built-in binaural functionality of the Web Audio API, which is derived from *IRCAM Listen*'s head-related transfer function (HRTF) database [10]. However, several studies have shown that non-individualized HRTFs yield inconsistent results across listeners in terms of localization accuracy [11]. Although the Web Audio API does not currently support the use of custom HRTFs, recent studies have shown that it can be extended to allow users to upload individualized HRTFs [10, 9].

2.1.2 Virtual Acoustic Environments

Studies on virtual acoustic environments (VAEs) investigate the modeling of sound propagation in virtual environments through source, transmission, and listener modeling [12]. In the 1990s, Huopanemi et al. [13] developed DIVA Virtual Audio Reality System as a real-time virtual audiovisual performance tool with both hardware and software components. The system used MIDI messages to move virtual instruments in space using binaural rendering.

A commercial application of VAEs is the simulation of room acoustics for acoustic treatment purposes. In such applications, a specialized software allows the users to load architectural models and surface properties to simulate propagation characteristics of sound within a given space, such as a concert hall, theatre, office, or a restaurant. In a basic auralization (or sound rendering) pipeline used in VAEs, the acoustic model of a virtual environment is used to filter an audio signal to create an auditory display in the form of a spatialized signal [14, 15]. While previous projects have offered efficient methods for the rendering of virtual acoustic environments [16, 17, 18], it remains a challenging task to compute a high-density sonic environment with acoustic modelling, as the computational load depends linearly on the number of virtual sources [17].

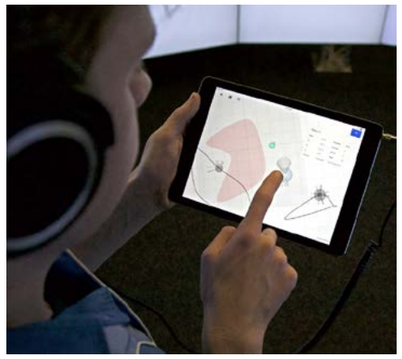


Figure 2. A user exploring the augmented reality in a CAVE system, while using a mobile device to edit the 3D sonic virtual reality he is hearing through headphones. The user is controlling the position of an object in lateral-view mode.

3. OVERVIEW OF THE SYSTEM

A system for the computational design of virtual soundscapes requires audio-to-visual representations. In digital audio workstations, a sound element is represented by a horizontal strip that extends over a timeline, where the user can edit a single sound element by cutting and pasting portions of this strip. Furthermore, multiple strips can be aligned vertically to create simultaneous sound elements. However, in the context of a virtual reality application, conceiving sound elements as spatial entities, as opposed to temporal artifacts, requires a different framework. To represent the different components of spatialized sound, we use visual elements— such as spheres, cones, splines and polygons— that are more applicable to the spatial composition of a sonic environment.

Based on the JavaScript library Three.js, our system utilizes a 3D visual scene, which the user can view at different angles to edit the layout of objects. However, manipulating and navigating an object-rich 3D scene using a 2D display can get complicated. Previous work has shown that, in such cases, using separate views with limited degrees of freedom is faster than single-view controls with axis handles [19]. Accordingly, in our system, the 2D bird's-eye view allows the user to manipulate the position of components on the lateral plane, while the 3D perspective view is exclusively used to control the height of the objects or trajectory control points.

We provide a unified environment for designing both open-space sonic environments and the sound objects contained within them. We combined a multiple-scale design [20] with a dual-mode user interface [21], which improves the precision at which the user can control the various elements of the virtual soundscape, from sound cones to sound objects to sound fields. We also utilized dynamic attribute windows to offer parametric control over properties that are normally controlled via mouse or touch interactions. This enables a two-way interaction between abstract properties and the virtual environment in a combined design space [22], which is used in information-rich virtual environments such as ours.

Furthermore, our system allows the user to simultaneously design and explore a virtual sound field. In modern game engines, the editing and the simulation phases are often separated due to performance constraints. However, since our underlying system is designed to maintain an audio environment, which is computationally less demanding than graphics-based applications, editing and navigation can be performed concurrently.

Finally, we offer an amalgamation of virtual and augmented reality experiences for the user. Given the ability of our system to function both on desktop and tablet computers, the user of an augmented reality implementation can manipulate the virtual environment using a mobile device while exploring the physical space onto which a virtual soundscape is superimposed, as seen in Fig. 2.

4. SOUND FIELD

The *sound field* is the sonic canvas onto which the user can place a variety of components, such as sound objects and sound zones. In the default state, the sound field is represented by a 2D overhead-view of an infinite plane. The user can zoom in and out of the sound field and pan the visible area. Furthermore, the sound field can be tilted and rotated. Whenever the user interacts with the sound field to add a new sound object, zone or trajectory, the view automatically switches to the bird's-eye view to allow for object placement. The user can then switch to the perspective view by clicking the view indicator on the bottom right corner of the interface. A global mute button allows the user to turn off the entire audio output. This makes it possible to make offline editions to the sound field. Furthermore, with dedicated icons found adjacent to the mute button, the user can save and load system states to restore a previously composed sound field.

4.1 Navigating the Interactive Virtual Soundscape

The user can explore the virtual sonic environment via one of two modalities, or a combination of both. In *virtual navigation*, a stationary user is equipped with a headphone connected to the device running the system. Using keyboard controls, the user can travel within the sound field

virtually. In *augmented navigation*, the user moves physically within a room that is equipped with a motion-tracking system. User's gaze direction is broadcasted to the system via OSC to update the position and the orientation of the Web Audio's *Listener Node*, which effectively controls the binaural rendering of the auditory scene based on the user's movements.

In augmented reality applications of our system, the user can define a sub-plane within the sound field to demarcate the region visible to the motion-tracking system. The demarcated region is represented by a gray translucent polygon on the sound field. The users can adapt the room overlay to the particular room they are in by mapping the vertices of this polygon to the virtual positions tracked when they are standing at the corners of the room. Sound components can be placed inside or outside the boundaries of the room.

5. SOUND OBJECTS

5.1 Multi-cone implementation

In modern game engines, users can populate a scene with a variety of visual objects. These objects range from built-in assets to 3D models designed with third-party software. Sound assets are phantom objects that define position and, when available, orientation for sound files that are to be played back in the scene. Sound assets can be affixed to visual objects to create the illusion of a sound originating from these objects. Directionality in game audio can be achieved using sound cones. A common implementation for this consists of two cones [7]: an inner cone plays back the original sound file, which becomes audible when the user's position falls within the projection field of the cone. An outer cone, which is often larger, defines an extended region in which the user hears a attenuated version of the same file. This avoids unnatural transitions in sound levels, and allows a directional sound object to fade in and out of the audible space.

However, sound producing events in nature are much more complex. Parts of a single resonating body can produce sounds with different directionality, spread, and throw characteristics. With a traditional sound cone implementation, the user can generate multiple cones and affix them to the same point to emulate this behavior, but from a UI perspective, this quickly gets cumbersome to design and maintain. In our system, we have implemented a multi-cone sound object that allows the user to easily attach an arbitrary number of right circular cones to a single object, and manipulate them.

5.2 Interaction

After pressing the *plus* icon on the top right corner of the UI, the user can click anywhere in the sound field to place a new sound object. The default object is an ear-level [6] omnidirectional point source represented by a translucent sphere on the sound field.

Creating a new object, or selecting an existing object, brings up an interactive close-up view, as seen in Fig. 3, as well as an attribute window on the top right region of the screen. The sound field view remains unchanged providing the user contextual control over the object that is being edited in the close-up window. The close-up view allows the user to add or remove sound cones and position them at different longitude and latitude values. Interacting with a cone brings up a secondary attribute window for local parameters, where the user can attach a sound file or an audio stream to a cone, as well as control the cone's base radius and lateral height values. The base radius controls the projective spread of a sound file within the sound field, while the height of a cone determines its volume. These attributes effectively determine the spatial reach of a particular sound cone. The secondary attribute window also provides parametric control over longitude and latitude values. Each object can be duplicated with all of its attributes. A global volume control allows the user to change the overall volume of an object, which is represented by the radius of the translucent sphere.

[6] Ear-level is represented by the default position of the audio context listener object on the Y-axis.

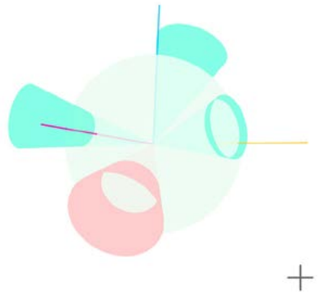


Figure 3. A screenshot of the object close up view displaying a sound object with four cones. The cone in red is currently being interacted with.

5.3 Trajectories

The user can attach arbitrarily drawn motion trajectories to each sound object. If the start and stop positions of a trajectory drawing are in close proximity, the system interpolates between these points to form a closed-loop trajectory. Once the action is completed the object will begin to loop this trajectory using either back-and-forth or circular motion depending on whether the trajectory is closed or not. Once a trajectory has been defined, a trajectory attribute window allow the user to pause, play, change motion speed in either direction or delete the trajectory. A resolution attribute allows the user the change the number of control points that define the polynomial segments of a trajectory curve. Once the user clicks on an object or its trajectory, these control points become visible and can be repositioned in 3D.

6. SOUND ZONES

For ambient sounds or sounds that are to be perceived as originating from the listener, we have implemented the sound zone component, which demarcates areas of non-directional and omnipresent sounds. Once the user walks into a sound zone, they will hear the source file attached to the zone without distance or localization cues.

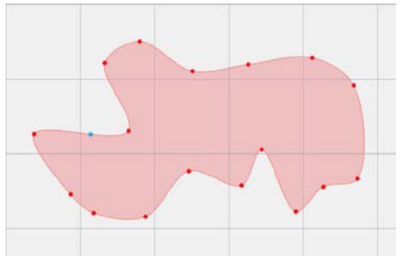


Figure 4. A screenshot of a sound zone that is being edited in the bird's-eye view mode. The user is about to add a new control point at the location highlighted with the blue dot.

6.1 Interaction

After clicking the *plus* icon on the top right corner, the user can draw a zone of arbitrary size and shape within the sound field with a click-and-drag action. Once the action is completed, the system generates a closed spline curve by interpolating between action start and stop positions. When a new zone is drawn, or when an existing zone is selected, a window appears on the top right region of the screen to display zone attributes, which include audio source, volume, scale, and rotation. An existing zone can be reshaped by adding new control points or moving the existing ones, as seen in Fig. 4.

7. APPLICATIONS

The ease of use, detail of control, and the unified editing and navigation modes provided by our system not only improve upon existing applications but also open up new creative and practical possibilities.

Interactive virtual soundscapes have many applications ranging from artistic practice to data sonification. As a compositional tool, our system constitutes a platform to create works consisting of sounds that act as spatial entities rather than events that are part of a temporal progression, which is often emphasized in modern digital audio workstations. Our system allows the composer to visualize sound sources located in space, and therefore have a better grasp of the spatial configuration of separate sound objects. Using our built-in objects, the composer can create complex sound morphologies, and layer a multitude of objects to explore spatially emergent sonic characteristics. Furthermore, the real-time design features of our system make it possible to use the system in concert situations, where the artist's construction of a virtual soundscape becomes a part of the performance. Furthermore, besides these uses that are intended for the artist, the listeners can also use our system to create casual spatial listening experiences.

Our system can also be utilized in sound pedagogy. Ear cleaning exercises, first proposed by R. Murray Schafer [23], aim at improving people's awareness of not only their immediate sonic environments, but also the precision with which they can listen to their surroundings. Ear cleaning exercises focusing on dynamic, spectral and spatial characteristics of environmental sounds can be administered using our system. Multi-participant exercises can be conducted using the augmented reality system. Furthermore, new ear cleaning exercises, such as re-spatializing of real-time audio input in the virtual soundscape, can be envisioned.

While our system relies on basic and widely-adopted mouse and touch interactions, it also affords a parametric control of object, zone and sound field attributes. This allows it to be utilized as a sonification tool for scientific applications, where researchers can rapidly construct detailed and accurate auditory scenes. Furthermore, since it can receive and transmit OSC data, our system can be interfaced with other software. This allows the control of sound objects via external controllers or data sets, but also enables the system to broadcast sound field data to other applications with OSC capabilities, such as Processing, openFrameworks and Unity.

Our system can also be used as an on-sight sketching tool by landscape architects to simulate the sonic characteristics of open-air environments. By mapping the target location on our sound field, the architect can easily construct a virtual environment with sound producing events within both the target location and the area surrounding it.

8. FUTURE WORK AND CONCLUSIONS

In the near future, we plan to implement 3D objects that enable sound occlusion. This implementation will allow the artist to draw non-sounding objects in arbitrary shapes that affect the propagation of sounds around them. Furthermore, although velocity functions that are used to achieve doppler effects has been deprecated in the recent version of the Web Audio API, we plan to add this feature to better simulate objects with motion trajectories. We also plan to improve the sound zone implementation with gradient volume characteristics. Similar to radial and linear gradient fill tools found in graphics editors, this feature will allow the user to create sound zones with gradually evolving amplitude characteristics. Additionally, we plan to implement features that will facilitate rich mixed reality applications. For instance, incorporating a video stream from the tablet camera will allow the user to superimpose a visual representation of the sound field onto a live video of the room they are exploring with a tablet.

In this paper, we introduced a novel system to design and control interactive virtual soundscapes. Our system provides an easy-to-use environment to construct highly-detailed scenes with components that are specialized for audio. It offers such features as simultaneous editing and navigation, web-based cross-platform operation on mobile and desktop devices, the ability to compose complex sound objects and sound zones with dynamic attributes that can be controlled parametrically using secondary attribute windows, and multiple views to simplify 3D navigation. As a result, our system provides new creative and practical possibilities for composing and experiencing sonic virtual environments.

9. REFERENCES

[1] R. Zvonar, "A History of Spatial Music: Historical Antecedents from Renaissance Antiphony to Strings in the Wings," *eContact*, vol. 7, no. 4, 2005.

[2] D. R. Begault, *3-D Sound for Virtual Reality and Multimedia*. San Diego, CA, USA: Academic Press Professional, Inc., 1994.

[3] F. Grani, S. Serafin, F. Argelaguet, V. Gouranton, M. Badawi, R. Gaugne, and A. Lécuyer, "Audio-visual Attractors for Capturing Attention to the Screens when Walking in CAVE Systems," in *IEEE VR Workshop on Sonic Interaction in Virtual Environments (SIVE)*, 2014, pp. 75–76.

[4] A. Çamcı, Z. Özcan, and D. Pehlevan, "Interactive Virtual Soundscapes: A Research Report," in *Proceedings of the 41st International Computer Music Conference*, 2015, pp. 163–169.

[5] C. Miyama, G. Dipper, and L. Brümmer, "Zirkonium Mk III: A Toolkit for Spatial Composition," *Journal of the Japanese Society for Sonic Arts*, vol. 7, no. 3, pp. 54–59.

[6] M. Geier and S. Spors, "Spatial Audio with the SoundScape Renderer," in *27th Tonmeistertagung–VDT International Convention*, 2012.

[7] P. Adenot and R. Toy. (2016) Web Audio API. [Online]. Available: http://webaudio.github.io/web-audio-api/.

[8] M. Rossignol, G. Lafay, M. Lagrange, and N. Misdariis, "SimScene: a Web-based Acoustic Scenes Simulator," in *Proceedings of the 1st Web Audio Conference*, January 2015.

[9] C. Pike, P. Taylour, and F. Melchior, "Delivering Object-Based 3D Audio Using The Web Audio API And The Audio Definition Model," in *Proceedings of the 1st Web Audio Conference*, January 2015.

[10] T. Carpentier, "Binaural Synthesis with the Web Audio API," in *Proceedings of the 1st Web Audio Conference*, January 2015.

[11] S. Zhao, R. Rogowski, R. Johnson, and D. L. Jones, "3D Binaural Audio Capture and Reproduction Using a Miniature Microphone Array," in *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx)*, 2012, pp. 151–154.

[12] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen, "Creating Interactive Virtual Acoustic Environments," *J. Audio Eng. Soc*, vol. 47, no. 9, pp. 675–705, 1999.

[13] J. Huopaniemi, L. Savioja, and T. Takala, "DIVA Virtual Audio Reality System," in *Proceedings of International Conference on Auditory Display (ICAD)*, November 1996, pp. 111–116.

[14] T. Funkhouser, J. M. Jot, and N. Tsingos, ""Sounds good to me!"–Computational Sound for Graphics, Virtual Reality, and Interactive Systems," *ACM SIGGRAPH Course Notes*, pp. 1–43, 2002.

[15] T. Takala and J. Hahn, "Sound Rendering," *SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 211–220, Jul. 1992.

[16] R. Mehra, A. Rungta, A. Golas, M. Lin, and D. Manocha, "WAVE: Interactive Wave-based Sound Propagation for Virtual Environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 4, pp. 434–442, 2015.

[17] M.-V. Laitinen, T. Pihlajamäki, C. Erkut, and V. Pulkki, "Parametric Time-frequency Representation of Spatial Sound in Virtual Worlds," *ACM Transactions on Applied Perception (TAP)*, vol. 9, no. 2, p. 8, 2012.

[18] T. Yiyu, Y. Inoguchi, E. Sugawara, M. Otani, Y. Iwaya, Y. Sato, H. Matsuoka, and T. Tsuchiya, "A Real-time Sound Field Renderer Based on Digital Huygens' Model," *Journal of Sound and Vibration*, vol. 330, no. 17, pp. 4302 – 4312, 2011.

[19] J.-Y. Oh and W. Stuerzlinger, "Moving Objects with 2D Input Devices in CAD Systems and Desktop Virtual Environments," in *Proceedings of Graphics Interface 2005*, 2005, pp. 195–202.

[20] B. B. Bederson, J. D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas, "PAD++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics," *Journal of Visual Languages and Computing*, vol. 7, pp. 3–31, 1995.

[21] J. Jankowski and S. Decker, "A Dual-mode User Interface for Accessing 3D Content on the World Wide Web," in *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 1047–1056.

[22] D. A. Bowman, C. North, J. Chen, N. F. Polys, P. S. Pyla, and U. Yilmaz, "Information-rich Virtual Environments: Theory, Tools, and Research Agenda," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. ACM, 2003, pp. 81–90.

[23] R. M. Schafer, *Ear Cleaning, Notes for an Experimental Music Course*. Toronto, CA: Clark & Cruickshank, 1969.

The Paradox of Random Order

Jeyong Jung
Institute of Sonology,
Royal Conservatoire,
Juliana van Stolberglaan 1,
Den Haag, The Netherlands
jey.noise@gmail.com

Peter Pabon
Institute of Sonology,
Royal Conservatoire,
Juliana van Stolberglaan 1,
Den Haag, The Netherlands
pabon@koncon.nl

Graham Flett
Institute of Sonology,
Royal Conservatoire,
Juliana van Stolberglaan 1,
Den Haag, The Netherlands
gflett@koncon.nl

ABSTRACT

This paper seeks to describe a system of period-by-period synthesis, one that uses discrete random function distributions. These apply both to micro and macro control signals for sound synthesis. Additionally, the system innately generates sound in a "noisy-timbre" and has been used to realize musical compositions in both real-time and non-real-time environments.

1. INTRODUCTION

In signal processing, the proposition of reordering a population of *independent and identically distributed (IID)* random numbers presents an inevitable paradox. This is evident from the fact that, given any set of IID random values, randomizing sequential order hardly affects the spectrum of a resulting wave. If the boundary of this discourse is expanded, to include the sequential nature of musical activities, then the original proposition is one made in vain—as the more fundamental issue resides in who is making the sequence in question. Techniques introduced in this paper can thus be understood as ways for increasing controllability in creating random sequences, primarily done by meticulously defining a *probability distribution function (PDF)* of control data. Furthermore, these utilize the paradox of random order in sound production. As such, concatenated waves form sound materials, then a composer finds synchronic relations among the sound materials.

2. NOISE TRANSFORMATION

For decades, this technique of redistributing discrete random data according to a predefined PDF has engaged in a certain fashion of computer music. Noise transformation suggests different ways of achieving this goal. Moreover, a notable mathematical feature of noise transformation is that it can involve a two-to-one mapping between the domain (incoming random numbers) and the range (redistributed random numbers).

2.1 Noise Transforms

When a population of N uniform-distributed random samples S in the range $0 \le S < 1$ is examined sequentially through three boundaries given as (1) $0 \le S <$ A, (2) A $\le S <$ B, (3) B $\le S < 1$ and the premise "$0 \le$ A $<$ B ≤ 1"[1] is satisfied, "then approximately N · (B – A) samples will fall"[1] into the second boundary. In these *noise transforms*[1], a finite number of boundaries are held within uniformed distributed probabilities, these then scale down a continuous population distribution until reaching a discrete one. By postulating six boundaries such as (1) $0 \le S < 1/21$, (2) $1/21 \le S < 3/21$, (3) $3/21 \le S < 6/21$, (4) $6/21 \le S < 10/21$, (5) $10/21 \le S < 15/21$, (6) $15/21 \le S < 1$, that are mapped to a *sample space*—the range corresponds to given boundaries—such as {0, 1/5, 2/5, 3/5, 4/5, 1} one-to-one, the graph (a) in Figure 1 can be made. Also, we can inversely find the given probability density values from the 6 boundaries given: {1, 2, 3, 4, 5, 6}, and the resolution of the sample space in arithmetic progression from the length of the PDF list.

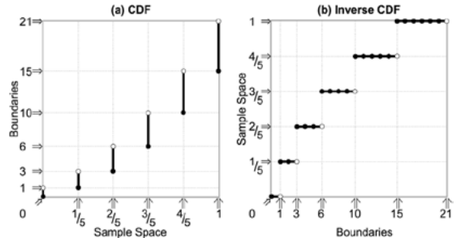


Figure 1. A CDF and an Inverse CDF.

2.2 CDF, Inverse CDF

Cumulative density (or distribution) function *(CDF)* and *inverse*-CDF (also called *quantile function*) are mapping functions used to generate a desired PDF. A feature of CDF is that a uniform chance—appearing within the sample space along the x-coordinate—can reveal a different density along the y-axis.

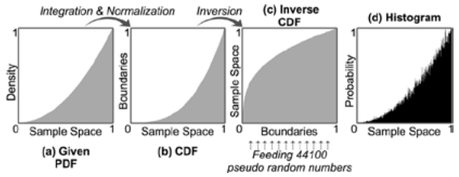


Figure 2. (a) $y=x^2$, (b) $y=x^3$, (c) $y=x^{1/3}$, and (d) "histogram of redistributed random numbers" are shown.

Thus we cannot feed a number through the y-axis of a function; instead using a CDF for mapping boundaries to

a sample space once again needs a searching process similar to the noise transforms. On the other hand, through an inverse CDF, a uniform chance—appearing within the boundaries along x-axis—will immediately find the corresponding samples along the y-coordinate.

2.3 Discretizing and Restoring an Inverse CDF

An inverse CDF elegantly simplifies the mapping in redistributing random samples. As it was mentioned, the PDF of noise transformation is prescribed by a list; naturally, the CDF and an inverse CDF derived from the given list will also become lists in their own right.

2.3.1 Discretizing an Inverse CDF

In case of (b) in Figure 1, the greatest common denominator among the values of the given PDF list is 1. Hence, the derived inverse CDF list can precisely express the density per each sample along the x-axis by having only 21 points. Consequently, when a given PDF list includes irrational numbers, and the same method of deriving (b) is used, an inverse CDF list becomes incredibly long.

Noise transformation seen in Figure 3 shows how a pseudo inverse CDF list can be derived (an example which does not need as many slots). In (a), the density of the 6 virtual axes—measuring the depth along the x-axis—are very poorly defined. Given this, the set of dotted lines shown in (b) of Figure 3 are greatly dissimilar to the ideal inverse CDF. Nevertheless, the dotted lines are useful for estimating which boundary in the CDF list a given sample will fall into.

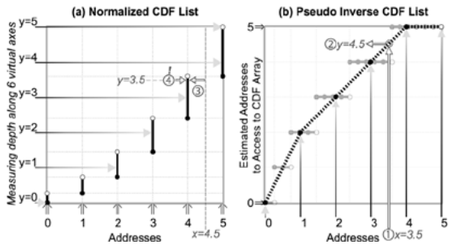


Figure 3. (b) is a discretized inverse CDF.

2.3.2 Inverse Search

At this point it is now necessary to discuss how the appearance of the series of the synthesis processes ①~④ (Fig. 3) was arrived at. In (b) ① and ② a pair of x-y coordinates {3.5, 4.5} were made; following the inverse relation between a CDF and an inverse CDF, the x-y coordinates were inverted in order to search through (a) and then become {4.5, 3.5}. As ③ shows, the x-coordinate 4.5 was then rounded down until it became 4. After this, at ④, it can be seen that the y-coordinate 3.5 belongs to the fifth boundary within (a). With regards to the manner of the process in ④, the results can be seen as false, however, there is a useful principle evident here from the statistical data. Due to the fact that negative probability-density-values cannot exist, every CDF is in turn a monotonically increasing function. If the boundary that is currently being compared with the given y-coordinate is in *above* or *below* of the given y-coordinate, the x-coordinate consequently moves *backward* or *forward* by 1. Accordingly, when a very long list prescribes probability density values that are only nearby both ends of the sample space, such as {1, …very many zeros …, 1}, the searching method introduced in this section becomes rather inefficient.

2.4 Interpolating a Sample Space

By stretching the boundaries in an inverse CDF along the sample space, a discrete prescription turns into a continuous function. Each boundary in the two (b)s of Figure 4 and 5 appear at the maximal spread, positing 1 as a factor, along the y-axis allowed by noise transformation.

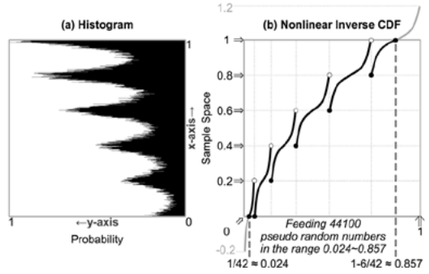


Figure 4. (b) consists of the unwrapped frequency warping functions.

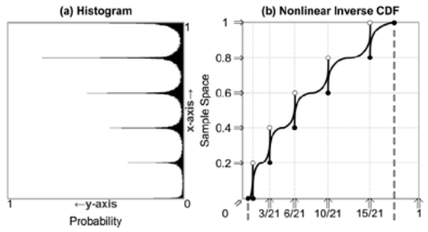


Figure 5. (b) contains the unwrapped $sin^{1/4}(x)$ functions.

In Figure 6, histograms were made by feeding samples in arithmetic progression into noise transformation. The PDFs appearing within were all prescribed by the same list, however, the prescription from this list was interpolated by using different modes. To derive (a) and (b), the inverse CDF needed to consists of the unwrapped frequency warping functions. Thus, (c) and (d) display the result of using an inverse CDF made up with imprecise inverse CDFs—based on the $sin^2(x)$ function. Additionally, the software version noise transformation also realized the interpolation of the three different states in real-time.

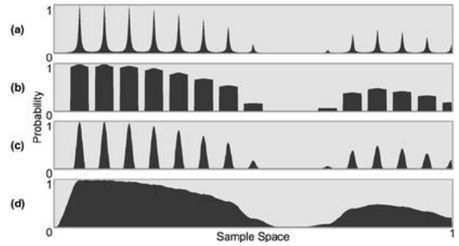


Figure 6. The spread factors given to noise transformation were, from (a) to (d), 1, 0.4, 0.25, and 1.

3. PERIOD-BY-PERIOD SYNTHESIS

Within the previous chapter, an individual sample value was the basic unit of the distribution measurements. Of course a random number stream made by noise transformation can be seen as sound wave, but due to the fact that the generated random numbers are IID, the number stream itself possibly possesses many sudden jumps that contain all possible frequency components, which results in a predominantly flat spectrum. The idea now is that the period-by-period synthesis can utilize random numbers as control data, involving an irregular rate of parameter determination.

3.1 Instructions

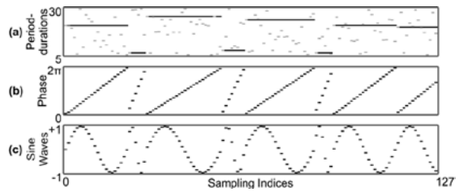


Figure 7. The period-by-period synthesis.

The black horizontal lines in (a) of Figure 7 express the chosen period-durations among random numbers (indicated as gray dots). These period durations are being held within the corresponding amount of sampling intervals. After this, these period durations turn into a concatenated set of single-period phase signals such as (b) in Figure 7. The possible period durations consist of all the real numbers within the desired duration range, therefore the slope of the phase signal should change at a point in between two consecutive sampling moments. When this occurs, within the following period, a constant offset—the phase increment within the "point of speed change" and the "following sampling moment"—would have to be applied to all phase values.

Period-by-period phase signal, when applied in random order, can be fed into any shaping function; furthermore, temporal synchronization between speed change and any other variables from other processes can also be made.

3.2 Jey-noise

As shown in (c) of Figure 7, the period-by-period phase signal can be fed into a sine function. These phenomena are called *jey-noise* and they realize a paradoxical situation where the random order, derived from constant probability distribution, results in various short-term spectra. It also makes the effect of applying arbitrary probability distribution upon control data more obvious within the spectrum.

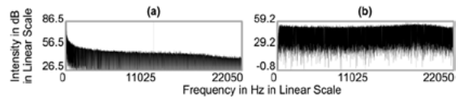


Figure 8. Two spectra of 10-second-long *jey-noise*s.

Examining the long-term spectra of *jey-noise* reveals that the spectral analysis (a) of Figure 8 has been derived from the result of allowing white noise to determine the period-durations of *jey-noise*. Moreover, by feeding white noise into a logarithmic function, the distribution of random period durations will change in such a way that shorter periods have a higher chance to appear and longer periods a lower chance, which furthermore equalizes the overall frequency distribution of a series of successive random period durations. As a result, the occurrence of period durations is redistributed with a more even/flat frequency distribution (see Fig. 8b).

3.3 Period-by-Period Spatial Wave Transformation

Gendee v. 0.1 is a polynomial wave-transformer that can rotate an exponential curve upon three axes (x, y and z) in 3D-space. As its name suggests, it is possible to draw an explicit connection between *gendee* and the sound synthesis techniques of Iannis Xenakis. Similar to the "Dynamic Stochastic Synthesis"[2] of Xenakis, *gendee* concatenates dynamically-changing waveforms.

By randomizing *gendee*, the probability distribution for each parameter is being prescribed by using the combination of the same exponential curve and the same wave-transformer. Furthermore, to derive a 2D-wave from a 3D-phenomenon, *gendee* removes the depth of the transformed exponential curve using orthogonal projection.

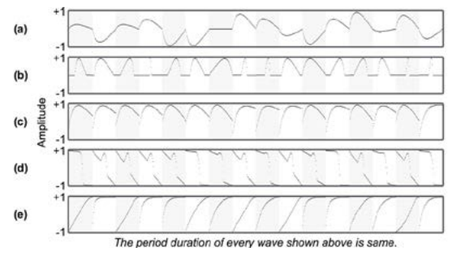


Figure 9. The sets of concatenated waves.

The graph (a) of Figure 9 shows how a tilted exponential curve is rotated along the x-axis. Similarly, the graphs (b) and (c) have been derived by rotating the same logarithmic curve (an inverted exponential curve) present upon the y and z axes. When a given curve rotates upon the y-axis, the width of the curve narrows and (d) demonstrates how the empty space that occurs along y-axis rotation can become full. In this way, if a period-by-period phase signal consists of exponential or logarithmic curves in various curvatures, a logarithmic curve is then transformed as shown in the graph (e).

4. APPLICATIONS

4.1 Using Jey-noise in a Voltage-Controlled Studio

Using broad-band *jey-noise* in a static state created source material for making a fixed-media piece entitled *Continuum*. The composition process was done entirely in a voltage-controlled studio at the Institute of Sonology.

This is similar to the composition process of the piece *Terminus*, where Gottfried Michael Koenig introduced the idea of "a mechanical derivation procedure." Koenig described such a procedure as being serial with respect to the way each derivation formed the basis for its successor.[3]

In composing *Continuum*, Jung's use of *jey-noise* transformed into the first generation materials, but then proceeds to ones made in advance of this generation, which are paradoxically turning them into their offspring. Jung then derives these materials out of the same lineage, organizing the sound materials into distinctive sections then concatenating sections in a sequence so that each section contrasts to adjacent ones. Similar to Koenig, who treated "the possible form-sections" as "closely linked … without having a goal-oriented relationship to each other"[3], Jung's *Continuum* also attempts to derive a diachronic narrative from within the relations of the coexisting materials.

4.2 Generative Bootstrapping

Bootstrapping is a process of deriving probability distribution of a population. The law of large numbers states that, when proceeding statistical trials, the greater the number of random samples present the greater the precision of any probability distribution analysis. However, generative bootstrapping pursues to elicit dynamics rather than precision. Seen within a limited number of trials, the analysis data of any population of random numbers correspondingly differs from the given PDF prescription by any degree. As a consequence, an incorrect analysis data becomes the new PDF list and the same recursion repeats itself. Such a feedback implements Koenig's concept of "serial" within statistical trials. Here too, an inexactitude of distribution analysis data paradoxically guarantees change.

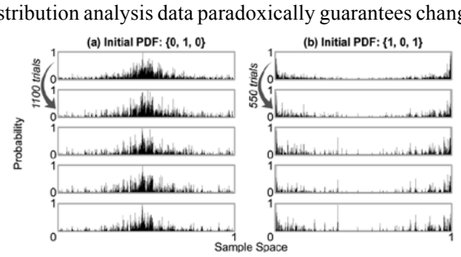


Figure 10. Each family originates from a distinctive list.

The number of trials determines how fast changes in probability data will be. Also, the "interpolation mode of the sample space of noise transformation" and the "resolution of histograms" affect the tendency of change in probability distribution data. This process and *jey-noise* were used in a live-performance entitled *Noise Gallery 1*[4].

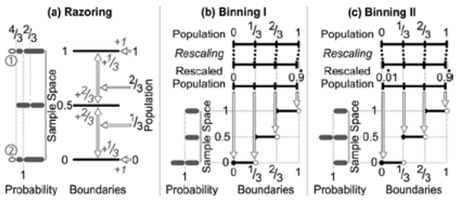


Figure 11. The PDF analysis methods are shown.

Let us now examine how a population was analyzed through the use of generative bootstrapping. The processes in Figure 11 show the analyses of the distribution of a population—including 4 samples {0, 1/3, 2/3, 1}—by the resolution of 3. So far in this paper, the process (a)—that can be understood as feeding samples into a CDF—was used in making histograms. In the case of (a), the direction of the descending samples run parallel to the boundaries, this means that only samples 0 and 1 will fall into the boundaries and the corresponding probability values will increase by 1. The other samples, 1/3 and 2/3, increase the probability values of adjacent boundaries by fractions. In case of the probability values on both ends, incrementing by fractions, only occurs from one side. ① and ② thus show how an incrementing 1/3 affects the probability values that correspond to the sample 0 and 1, which were doubly weighted.

When using *binning*—that uses an inverse CDF—in deriving histograms, the samples of the given population is always rescaled. The process (b) involves "$0 \le S < 1$ rescaling"—ensuring that the range of samples does not equate to exact 1—and results in the probability plot where the peak is on the leftmost sample. The process (c) involves the rescaling range of $0 < S < 1$ and re-positions the peak in the probability plot to be in the middle of the sample space. Furthermore, successive histograms made by using (c) will show an oscillation of density along the y-axis and an expansion along the sample space.

5. CONCLUSIONS

All processes so far discussed have already been used as part of modules integrated into either a live-performance system or voltage-controlled patches. Additionally, to elicit a variety of behaviors from this approach, it is conceivable to plan to expand the types and kinds of available modules. Also, introducing wave analysis methods will enrich the period-by-period synthesis technique. But at present there are still many possibilities of using noise transformation, especially for creating random number streams with incredibly long repetition period. Given this the authors of this paper admit to an overall curiosity to explore the technical concept of achieving consistency between the PDF prescriptions of noise transformation and the spectra of *jey-noise*. *Gendee* will furthermore be improved, soon able to rotate any given curve. Lastly, given that the histogram (d)—Figure 6—results in weak ripples on the envelope, it is foreseeable that a pseudo inverse CDF of the $sin^2(x)$ function will be used for future filter design.

Acknowledgments

The authors are indebted to the editing of Johan van Kreij.

6. REFERENCES

[1] C. Ames, "A Catalog of Statistical Distributions: Techniques for Random, Determinates and Chaotic Sequences." *Leonardo Music Journal* 1, no. 1 (1992): 55-70.

[2] S. Luque, "The Stochastic Synthesis of Iannis Xenakis." *Leonardo Music Journal* 19 (2009): 77-84.

[3] G. M. Koenig, "Genesis of Form in Technically Conditioned Environments." *Interface* 16, no. 3 (1987): 165-175.

[4] https://soundcloud.com/user-30688271

Recreating Gérard Grisey's Vortex Temporum with cage

Daniele Ghisi
STMS Lab (IRCAM, CNRS, UPMC)
Paris

Andrea Agostini
Conservatory 'G. Verdi'
Turin

Eric Maestri
LabEx GREAM, Université de Strasbourg
CIEREC, Université de Saint-Etienne

ABSTRACT

This paper shows how a significant portion of the first movement of Gérard Grisey's Vortex Temporum *can be implemented using a subset of the* cage *package for Max, aimed at representing and rendering* meta-scores*, i.e., musical scores whose notes or chords represent complex processes, rather than simple musical items. We shall also attempt a solid definition of meta-score, and describe the rationale and workings of the* cage.meta *system.*

1. INTRODUCTION

Vortex Temporum [1] is a highly formalized chamber music composition in three movements, written between 1994 and 1996 by French composer Gérard Grisey, and widely acclaimed as one of the masterpieces of the late twentieth century music. Among the literature discussing various aspects of the piece, a very detailed technical analysis of the musical formalization techniques and mechanisms it employs has been carried out by Jean-Luc Hervé [2]. Starting from this analysis, we decided to try to recreate a substantial portion of the first movement as a case study for the *cage* [3] and *bach* [4] systems, which are two Max packages devoted to computer-assisted composition and musical notation, with *bach* providing a low- and middle-level infrastructure, implementing dedicated data structures and musical notation editors, and *cage*, which depends on *bach*, implementing higher-level musical processes and helper tools. In particular, we based our work on one specific subset of the *cage* package, the *cage.meta* system, aimed at representing and rendering what we call *meta-scores*. By this term, we mean musical scores (in the wide sense of symbolic timelines), each of whose individual events represents a complex musical process, rather than a simple musical item such as a note. In this paper, we describe the *cage.meta* system, and how it has been used for recreating a portion of *Vortex Temporum*. [1]

[1] The patches described in this article can be downloaded at the url `http://data.bachproject.net/examples/icmc2016.zip`, and require Max 6.1.7 or higher (`http://cycling74.com`), bach v0.7.9 or higher, and cage v0.4 or higher (both downloadable at `http://www.bachproject.net`).

2. META-SCORES

2.1 Composition and hierarchies

Most musical compositions can be described, at least partially, as constituted of hierarchically-nested structures [5, 6]. According to the musical style and genre, as well as the features of each individual composition and the analytical approach, these structures can be more or less numerous, and the hierarchy can be deeper or shallower. They can represent a hierarchical subdivision of time, as in the classical sonata form in which three major structures (exposition, development, recapitulation) are in turn composed of smaller structures (e.g., the exposition is often described as being composed by a first thematic area, a modulating bridge, a second thematic area, and a conclusive section), which can be divided further (e.g., a thematic area can be divided into one or more themes, and their elaborations and connections), and further (e.g., a theme can be divided into phrases or periods), and further (e.g., a phrase can be divided into semiphrases), and further (e.g., a semiphrase can be divided into motivic cells); or they can represent musical objects that can overlap in time (e.g., in the exposition of a fugue one can discern a subject and a countersubject, which are meant to be played in counterpoint, one against the other); or both kinds of structure can be discerned in the same composition, at different levels of the hierarchy. Moreover, even in the case of a temporal hierarchy, different sections can overlap, and their boundaries can be blurred. This leads to scenarios whose degree of complexity can be quite high.

On the other hand, this kind of hierarchically-structured representation of a piece of music can be a powerful conceptual tool for musical composition. Composers often take advantage of nested structures, even in programmatically 'frugal' stylistic contexts such as rock music (it is hard to imagine a song like Nirvana's *Smells like Teen Spirit* not having been conceived with a clear architectural project of alternation of verse and chorus, each built upon the repetition of a four-bar melodic pattern, based in turn upon a double iteration of a two-bar harmonic pattern). In more complex scenarios, it may be debated whether structures which can be revealed by means of musical analysis are part of a rational, deliberate structural plan, or are the product of the composer's 'intuition', whatever meaning one wants to assign to the term. In general, though, it can be safe to assume that most musical compositions have been at least partially conceived in the rational terms of some kind of hierarchical structure.

Software systems for computer-aided composition mostly focus on the manipulation of basic musical elements, such

as notes and chords, or small-scale phenomena, such as melodic profiles. However, some of these systems provide tools for dealing with higher-level musical objects, treated as containers or generators of lower-level elements, one notable example being OpenMusic's *maquette* [7].

2.2 Hybrid scores

The concept of musical score is a crucial one. For Lydia Goehr, a score "must specify at least everything that is constitutive of a work" [8] and for Nelson Goodman it is "a notational language" in which "the compliants are typically performances" [9]. In short, the score is a prescriptive means instantiated by performances.

Since the 1950s, the role of musical notation has undergone a number of generalizations (including graphical scores, gestural scores, morphological scores in electroacoustic music). This evolution has been caused, in part, by the development of electronic music. Pierre Schaeffer defined two kinds of score: a descriptive one and an operational one [10]; Charles Seeger proposed the opposition between prescriptive and descriptive scores [11]. This differentiation is reinforced by the development of computer music: as a matter of fact, both computer programming and composing are mediated through notation [12]. Composers use scores to sketch musical ideas, formalize them into a script and communicate them to performers; computer programmers, on the other hand, mostly use symbol-based formal languages to convey instructions to a computer. In both cases, notation is prescription. Both musical notation and programming are systems of prescription of actions, specifically in the case of musicians that must activate a vibrating body, and tasks, in the case of computers, that activate the vibrating body of the loudspeaker mediating the intentionality of the musician: the combination between the two defines a *hybrid* dimension of musical scores as partially suggested by Simon Emmerson [13] and by Andrew Sorensen and Henry Gardner [14].

Hybrid scores have a twofold meaning: on the one hand, they are targeted to performers, to whom they prescribe actions which are typically, although not exclusively, aimed at producing sound; on the other hand, they are targeted to computers ('digital performers' [15]), to which, through information encoded in a programming language, they prescribe the production of sound or symbols, or even more complex tasks. In particular, hybrid scores are capable of prescribing (and hence embedding) other hybrid scores within themselves, which makes them very suitable to represent and process abstract, nested musical structures. Incidentally, such hybrid scores also contribute to the narrowing of the gap between scores and instruments [12].

Within this conceptual framework, we use the term *meta-score* to define a hybrid score whose components are not elementary notational items (typically, single notes or chords), but rather processes which can be further prescribed and described as scores in their own terms. So to speak, we might say that a meta-score is a score *of* scores (i.e., a score containing other scores), or a score *for* scores (i.e., a score containing instructions to build other scores), or a score *about* scores (i.e., a score containing descriptions of other scores), hence expanding the fundamental ideas described in [16].

3. CAGE.META

3.1 The basic idea

From the very beginning of the development of the *bach* library, we were encouraged to develop a system for dealing with meta-scores within the real-time framework of Max. After careful consideration, we came to the conclusion that we would not want to implement a dedicated graphical editor and paradigm (which is what the maquette is, within the OpenMusic environment), but rather to devise a design pattern allowing the usual *bach* notation editors/sequencers (the *bach.roll* and *bach.score* objects, respectively implementing time in a non-measured, proportional fashion, and in a measured fashion, with tempi, meters, traditional note values and rests) to be used for representing meta-scores, rather than scores proper. When we had the opportunity to develop the *cage* library [3], we decided to include in it two modules devoted to this specific problem, and constituting one of the various subsets of the library itself, namely the *cage.meta* subset.

The choice of extending to meta-scores the concepts and tools used for representing traditional scores is motivated by the observation that, somehow, there is no clear boundary between traditional and meta-scores. In fact, more often than not, symbols in any traditional score refer to extremely complex processes and activities, be it the skillful control of the friction of a bow on a string, or the triggering of complex sets of envelopes to be applied to the inputs and outputs of a bank of oscillators. Moreover, in historical musical practices, there exist specific synthetic notations representing complex musical figures, such as trills, mordents, arpeggi, gruppetti and other ornamentation marks, or — even more specifically — the numbers and symbols of figured bass. By not striking a dividing line between scores and meta-scores we aim to focus on the similarities, and the continuum, between the two, rather than on the differences. At the same time, we feel that a graphical interface based upon the traditional notational paradigm can be perceived as more 'neutral' than a custom one, and as such is less likely to suggest specific compositional approaches or practices, and more inviting to be bent to each composer's creative needs.

The basic idea behind *cage.meta* relies upon the fact that scores contained in *bach.roll* or *bach.score* objects are hybrid scores, as each of their notes can be seen as a container of various, heterogeneous parameters: a small, standard set of basic, required data which define the note itself in traditional terms (position in time, expressed in milliseconds in *bach.roll*, in bars and beats in *bach.score*; duration, expressed in the same respective units; pitch; and MIDI velocity), and an optional, user-definable combination of other associated data belonging to a wide array of types (numbers, text, breakpoint functions, lists of file names, and more), contained in specialized data structures called *slots*, with the only constraint that associations between individual slots and their data types are global with respect to the score (e.g., the first slot of all the notes of a score might be a container of a number, or the sixth slot a container of a list of file names). This restriction is meaningful in that it encourages establishing a correspondance between each slot and one parameter (or one coherent and well-defined set of parameters) to be controlled through one slot. On the other hand, when more flexibility is needed, text and list [2] slots may act as dynamically-typed data containers, as they can contain any combination of numbers, symbols and sublists. In any case, the actual data contained in each note, although constrained in type by the global association, are fully independent from all the other notes in the score. Slot data can be edited both graphically and algorithmically, by means of messages sent to the containing *bach.roll* or *bach.score* object, and queried at any time. Moreover, they are always returned at play time. This means that when a note with, for example, a breakpoint function in one of its own slots is encountered during playback, all its associated data are output from one dedicated outlet of the object containing the note itself, including its onset, pitch, velocity, duration, and all the points and slopes of the breakpoint function. These points and slopes can be used, for example, to control the amplitude envelope of the synthesizer responsible to render the score into audio. Indeed, in a typical scenario, slot data contain real-time parameters for DSP, but their scope and potential is much wider.

In the *cage.meta* system, each event of a meta-score is represented as a note (or possibly a chord, as discussed further) whose first slot contains the name of a Max patcher file implementing the process associated to the event itself: we shall say that the note *refers* to said patcher. At initialization time, the patchers referred to by all the notes of the score are loaded and individually initialized. At play time, when a note is met, all its parameters and slot data are passed to the patcher it refers to. Although this is not enforced in any way, the idea is that the patcher itself will activate its intended behavior according to these parameters when they are received. Because the duration of a note is passed as one of its parameters, it is possible for the activated process to regulate its own duration according to it — but, once again, this is not enforced by any means, and it is possible to implement processes whose duration is fixed, or depends on other parameters. The same goes for the pitches and the MIDI velocities: the fact that they are passed to the process does not mean that the process itself must use them in any traditional, literal way — in fact, it can as well ignore them altogether.

In practice, all this is achieved by means of two modules, named *cage.meta.engine* and *cage.meta.header*.

3.2 Building the infrastructure

A meta-score system is built in two distinct phases. The first phase is creating the patchers implementing the processes that will be associated to the meta-score events. Each patcher must contain one *cage.meta.header* module: at play time, parameters from the referring note will not be passed to these patchers through inlets, but by the third or fourth outlet of *cage.meta.header*, according to some settings which will be described in detail below. [3] The fact that one single inlet is passing all the parameters to the patcher is not a limitation, thanks to the ability of the *llll* data structure of representing hierarchically-structured collections of data of arbitrary breadth and depth.

The second phase is setting up the meta-score system, constituted by a *bach.roll* or *bach.score* object (which we shall refer to as the 'score' object from now on) connected to a *cage.meta.engine* object in a 'loopback' configuration, such that the two inlets of *cage.meta.engine* are connected respectively to the leftmost outlet of the score object, from which all the score data can be output as one possibly large *llll*, and its one-but-rightmost outlet, from which the data of each note are output as the note itself is encountered at play time. Also, the first outlet of *cage.meta.engine* must be connected to the leftmost inlet of the score object: in this way, *cage.meta.engine* can perform queries on it and set some of its parameters if required. Indeed, if the *format* message is sent to a *cage.meta.engine* module connected as described to a score object, the first slot of the latter is initialized to the 'file list' data type (although only one file name will be used), and the second slot to the 'integer' data type (so as to store the aforementioned optional instance number). Finally, a different *bach.roll* or *bach.score* object (according to the type of the meta-score object) can optionally be connected to the second outlet of *cage.meta.engine*, so as to collect a rendered score, according to a different usage paradigm which will be discussed below.

3.3 Implementing the meta-score

Now it is possible to write the actual meta-score: as said before, the first slot of each note will contain a reference to the patcher file implementing its process. This poses a problem: what happens if the same process must be triggered by more than one note, over the course of the meta-score? One might want each run of the process to depend on the result of the previous one, or to be completely independent from it. Moreover, polyphony must be taken into account: what happens if one note triggers a process, and before that process ends (which may or may not coincide with the actual end of the note as it is written) another note triggers the same process? Most often, one will want another independent execution of the same process to start, but it is also possible that the process itself is able to manage its own polyphony, in which case the same copy of the process that was already running should receive a new activation message. All these possible scenarios are managed through the optional assignment of an instance number to each note. Two notes referring to the same file with different instance numbers will activate two distinct instances of the same patcher; if the instance numbers are the same, the same instance of the patcher will be activated. If no instance number is provided, the note will activate the so-called 'global instance' of the patch, which is the same for all the notes without an instance number, and different from all the numbered instances. As a helper tool,

[2] The ubiquitous *bach* data type is a tree structure expressed as a nested list with sublists delimited by pairs of parentheses, and called *llll* (an initialism for Lisp-like linked list, hinting at the superficial similarity between the two representations). The *llll* can be somehow seen as a simpler and more general alternative to the Max *dictionary* data structure, without the obligation of associating symbolic keys to data. It should be remarked that the root level of an *llll*, as opposed to the Lisp list, is not enclosed by a pair of parentheses: as such, a flat *llll* is essentially equivalent to a regular Max message, thus allowing seamless exchange of data between standard Max objects and *bach* objects [4].

[3] For the sake of completeness, we will mention the fact that the first outlet of *cage.meta.header* should be connected to a *thispatcher* object, necessary to perform some ancillary operations such as programmatically hiding or showing the patcher window, or disposing the patcher itself; and the second outlet reports the patcher instance number, a concept which will be discussed shortly.

when *cage.meta.engine* receives the *autoinstances* message, it will automatically assign an instance number to each note referring to a patcher file, in such a way that instance numbers for each patch are minimized, with the constraint that overlapping notes (according to their own durations plus a 'release time' which can be globally set) referring to the same patcher will always receive different instance numbers — in a similar fashion to what happens with voice assignment in a polyphonic synthesizer. There is one more parameter influencing the choice of the copy of each patcher to be activated for one specific note: the engine name. Each instance of the *cage.meta.engine* module can be assigned an optional name, to be passed as the object box argument (i.e., one can type *cage.meta.engine foobar* in the object box, thus assigning the *foobar* name to *cage.meta.engine*). By assigning them different engine names, more than one *cage.meta* system can run simultaneously in one Max session, without interfering with each other. This naming also allows building nested *cage.meta* systems: a whole *cage.meta* system, composed of a meta-score, a *cage.meta.engine* module and a set of process patches, can be included in a process patch of another, higher-level *cage.meta* system, as long as the names of the *cage.meta.engine* modules are unique.

After the meta-score has been written, generated or loaded from disk, the *load* message can be sent to *cage.meta.engine*: this causes the score to be queried for all its file names and instance numbers, and loads each referred patch as many times as required by the different instance numbers found in the score. Immediately after having being loaded, each patch is initialized, that is, it is sent three identifiers: the engine name, the patcher's own file name, and its instance number. These three identifiers will be used at play time to route the parameters of each note to the correct instance of its referred patch only, while avoiding conflicts with other possible *cage.meta* systems running at the same time, in the same Max session. Furthermore, depending on *bach.roll*'s or *bach.score*'s attributes, markers and tempi can also be sent to all the patches at play time. Receiving notifications for markers can be useful if, for instance, one needs to adapt the behavior of a process to different sections of the score. As a convenient debug tool, selecting a note in the score editor and pressing the 'o' key causes the corresponding referred patch to be opened.

3.4 Playback and rendering

In principle, the outcome of a run of the meta-score is just data, which can be collected in any possible data structure, directed to any possible device or process, and to which any possible meaning can be assigned. Each process, as implemented in the corresponding patcher, receives data from *cage.meta.header* and produces a result which can be routed, for instance, to a MIDI device, or an audio output: but also, according to a less inherently real-time paradigm, to an audio buffer collecting algorithmically-generated audio samples; or to a new score object which will contain the 'rendering' of the meta-score. In particular, we deemed this last scenario to be so typical and peculiar that it deserved some kind of special treatment. More specifically, we expect most musical processes that users may want to implement with the *cage.meta* system to produce either a real-time audio, MIDI or OSC result, or a symbolic result (i.e., a score) to be built incrementally note by note, or chord by chord. As an example of the former case, each *cage.meta.header* patch could contain a synthesizer, generating a complex audio stream depending on the parameters of the associated note; in the latter case, each patch could produce a complex musical figure (e.g., an arpeggio) built according to the parameters of the associated notes, and meant to be transcribed into a final score resulting from the run of the whole meta-score. The latter case can be seen as a special case of the general one, but the complexity of setting up a system for the very basic purpose of generating a score starting from a meta-score prompted us to implement a specific mechanism allowing a process patcher to return to *cage.meta.header* one or more chords in response to a message coming from *cage.meta.header* itself.

More specifically, when the 'playback' attribute of *cage.meta.engine* is set to 1, events coming from the score object are passed to each process patch through the third outlet of *cage.meta.header*, and can be routed to any generic destination (real-time audio, MIDI, OSC, or anything else): for example, the synthesizer implemented in the process patch would set its own parameters according to the data received from the meta-score, activate itself and produce an audio signal to be directly fed to the audio output of Max.

On the other hand, when the 'render' attribute of *cage.meta.engine* is set to 1, events from the score object are passed to each process patch through the fourth and rightmost outlet of *cage.meta.header*, and one or more chords (that is, *llll*s featuring all the chords and note parameters, formatted in the *bach* syntax) can be returned to the second and rightmost inlet of the same *cage.meta.header* module, in a loopback configuration. [4] The *cage.meta.header* module then returns the received chords to its master *cage.meta.engine*, which formats them in such a way to allow an instance of the appropriate object, connected to its second outlet, to be populated with the score being rendered. All this is preceded by a sequence of formatting instructions sent to the destination *bach.roll* or *bach.score*, and generated only if the *render* attribute is on. If only one of the two mechanisms is implemented in the process patches, it is advisable not to activate the other, so as to avoid unnecessary and potentially expensive operations. At the end of the rendering process, the whole rendered score will be contained in the notation object connected to *cage.meta.engine*'s second outlet. So, for example, a patch building an arpeggio receives the parameters of note of the meta-score referring to it (and containing the parameters of the arpeggio, such as starting pitch, range and speed) from the fourth outlet of *cage.meta.header*, and returns the rendered arpeggio, as a sequence of notes, to the rightmost inlet of *cage.meta.header*. The notes of arpeggio are then sent by *cage.meta.header* to the master *cage.meta.engine*, which in turn formats them as messages for the *bach.roll* or *bach.score* object connected to its second outlet. Through this mechanism, this destination *bach.roll* or *bach.score* is incrementally filled with contents and eventually it will contain the result of the whole rendering process.

As a final note, it is possible to have both the *playback* and the *render* attributes set to 1, in which case each event in the meta-score will be routed to both the third and the fourth outlet of the corresponding *cage.meta.header* object, which in turn will expect to receive a rendered sequence of chords in its second inlet.

3.5 Running the system

When the 'play' message is sent to the score object, the playback and/or rendering of the meta-score begins: the score object starts outputting the notes it contains according to their very temporality, and *cage.meta.engine* and *cage.meta.header* cooperate to route the data associated to each note to the correct instance of the patcher the note itself refers to. Another possibility is sending the score object the 'play offline' message: in this case, the score object starts outputting the notes it contains in strict temporal order, but with the shortest possible delay between them, without respecting their actual onsets and duration. This is somehow analogous to the 'offline render' command that can be found in virtually any audio and MIDI sequencer. As hinted above, this is useful to trigger non-realtime rendering processes, such as, typically, the rendering of a score through the *lambda loop* of *cage.meta.header*, but also, for instance, the direct writing of samples in an audio buffer, or any other kind of batch operation.

3.6 Final considerations

Everything that was said so far implied that events are represented as data associated to notes of the meta-score, but a slightly different approach is possible. Both *bach.roll* and *bach.score* have a notion of chords as structures of notes sharing the same onset: indeed, single notes appearing in the scores are actually represented as one-note chords, and *bach.score*'s rests are noteless chords; on the other hand, it is possible for two distinct chords in the same voice of a *bach.roll* object to have the same onset. The key point here is that it is possible to choose whether *bach.roll* and *bach.score* should play back the score they contain in a note- or chord-wise fashion: of course, this choice is applicable to both the real-time and offline playback processes. When a single-note chord is encountered, there is no substantial difference between the two modes. A multi-note chord will be output as a sequence of individual messages in the former case, as a single message containing the data for all the concerned notes in the latter case. [5] By setting the playback to chord-wise, it is possible to define processes taking more than one note as their parameters (e.g., an arpeggiator, or a stochastic process based upon a pitch range). In this case, the patcher name and instance number should be assigned to one note per chord: if more than one note has these data assigned, it is undefined which one will be taken into account for initialization and playback.

It should also be remarked that each item (that is, each note, chord, measure, or voice) in a *bach.roll* or *bach.score* object can be individually muted or soloed. Through this feature, it is possible to control which events of a meta-score will be actually played back or rendered, thus making it easy to build partial, or alternative, versions of the same score. Moreover, although the *cage.meta* system has been designed with a musical usage in mind, its potential is actually wider: for example, it is easy to imagine it applied to generative video, or complex DMX lighting scenarios, or the automation of physical actuators, or, in general, any device or system implemented in, or controlled by, Max.

As a final note, this system also has some shortcomings. First of all, it relies on the synchronicity of Max *send* and *receive* objects across different patchers, which is not guaranteed if the 'Scheduler in Audio Interrupt' option of Max is active. Also, the playback fully relies on the sequencing system of *bach* editors, which output note data (including all slots) at once whenever the playhead reaches a note onset. Due to this, it is impossible to modify the parameters of a given note while it is playing, as modifications are taken into account only if they happen before the playhead reaches the note onset. For the same reason, scrubbing is not possible. Also, that there is no support for the *transport* system of Max. On the other hand, any sequencing feature supported by the *bach* objects (such as looping, jumping to an arbitrary position, and changing the play rate via a *setclock* object) will be perfectly usable within this system.

4. RECREATING *Vortex Temporum*

As a case study for the *cage.meta* system, we decided to recreate the first 81 measures (corresponding to numbers 1 to 20, according to the numbered sections marked in the score) of Gérard Grisey's *Vortex Temporum* in *bach* and *cage*, basing our work upon Jean-Luc Hervé's analysis [2].

The basic idea behind our exercise is abstraction: we aim at building and manipulating a meta-score featuring operative compositional elements, rather than pre-rendered symbolic processes. For instance, since the pitch choices in *Vortex Temporum* are strictly based upon a spectral paradigm, our meta-score will be solely composed of spectral fundamentals. [6] Every note in our meta-score is hence a fundamental for some process, and the indices of harmonics that are built upon it and used by the process are contained in the third slot of each note. We implemented both an off-line score rendering and a real-time rendition, the latter through an extremely simple synthesis process: for this reason, each note carries in a dedicated slot an amplitude envelope information, in the form of a breakpoint function.

Each note of the meta-score triggers one of three different processes: an arpeggio, an accented chord, or a long note. We shall now describe them in detail.

The patch `vt_arpeggiator.maxpat` is designed to generate all the arpeggio-like figures which characterize and identify *Vortex Temporum* from its very beginning. More specifically, the arpeggiator renders all the 16th-

[4] This kind of loopback configuration between the rightmost inlets and outlets of a module is a common design pattern in *bach* and *cage*, and because of its practical relation with Lisp's lambda functions it is called, conveniently if not entirely accurately, a *lambda loop* [4].

[5] As a technical note, it should be stressed that, although in most cases the note-wise playback of all the notes of multi-note chord will happen in one Max scheduler tick, there is no guarantee for that, as this depends on the number of same-onset notes, the other processes possibly running, and the user-definable scheduler settings. On the other hand, the same behavior would apply to the playback of multiple chords all having the same onset. So, the only reliable way of knowing at play time how notes are grouped into chords is by playing back the score chord-wise.

[6] It might be worth pointing out that all harmonic series used by Grisey in the part of *Vortex Temporum* that we reimplemented are stretched linearly in the pitch domain by a factor of $\pi/3$. This observation does not appear in [2], but it seems pertinent in our implementation.

notes figures, with the notable exception of the accented piano and string chords at the beginning of each numbered section: these figures have a different musical role (akin to attack transients), and will be rendered by a different module, which will be discussed further.

Besides the fundamental note and the list of harmonic indices, the arpeggiator also receives some additional content, contained in further slots of our meta-score: the duration of each rendered note in the arpeggio (it is, in our case, constantly 1/16); the number N of notes composing a single arpeggio period (for instance, for flute and clarinet at measure 1 we get $N = 8$, since the arpeggio loops after 8 notes); and the profile for the arpeggio period, as a breakpoint function representing time on the x axis, and the elements in the harmonics list on the y axis. The final point of this function should always coincide with the starting one (to comply with the looping).

Inside the arpeggiator patch, the arpeggio profile is sampled at N uniformly distributed points, each of which is then approximated to the nearest element in the list of the harmonics, which are uniformly distributed on the y axis, independently of their actual value, and subsequently converted into the pitch derived from the stretched harmonic series (see Fig. 1). All pitches are approximated to the quarter-tone grid, with the exception of piano notes, which are approximated to the semitonal grid. [7]

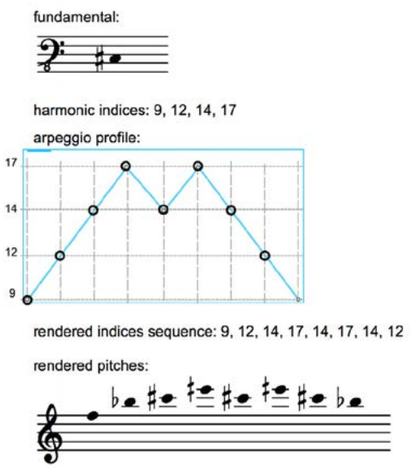


Figure 1. The conversion of the arpeggio profile for the flute (measure 1) into actual pitches. The harmonic indices, relative to the defined fundamental, are uniformly distributed on the y axis, and the profile is uniformly sampled on the x axis. The result is then snapped to the nearest harmonic index. The sequence is then rendered by retrieving the single harmonics of the stretched harmonic series built upon the fundamental.

During real-time playback, harmonics are then output by *bach.drip*, with the appropriate (uniform) time delay between them, depending on the current tempo and the duration of each rendered note. The audio rendering is performed by basic oscillators: the flute is rendered via a slightly overdriven sine tone; the clarinet via a triangular wave; the piano via a rectangular wave. These are of course completely arbitrary choices, only aimed at clearly illustrating the rendering mechanism.

Two accessory processes also need to be described. Firstly, during each of the short numbered sections, Grisey gradually filters out some notes from the arpeggi, replacing more and more of them with rests. This deletion only takes place in the flute and clarinet parts (also enabling the players to breathe between phrases), and it is roughly anticorrelated to dynamics. Since we do not have a model of the exact process employed by Grisey for performing these deletions, we decided to implement it probabilistically, by adding an 'existence probability' for each note, directly correlated to the instantaneous decibel value of the amplitude envelope: at 0dB, the existence probability will be 100%, while at -22dB the probability drops down to 50%. Secondly, starting from number 11, some bichords show up in the piano arpeggi. Bichords are always composed by neighbour harmonics in the harmonics list, thus preserving the arpeggio profile; hence it made sense for us to map a bichord to a half-integer snapping on the y axis of Fig. 1.

The second rendering module, implemented in the patch `vt_hits.maxpat`, deals with the previously mentioned accented chords at the beginning of each section. The offline rendering is trivial: each note is rendered as a chord whose pitches are the specified partials; no additional parameters are needed for this patch. During playback, each note is rendered as a white noise fed into a narrow-band resonant filter centered around the frequency of the note itself, and subsequently artificially reverberated.

The third rendering module, implemented in the patch `vt_tenuti.maxpat`, generates the long viola and cello notes (first appearing at number 3). The viola is rendered via a sawtooth oscillator, the cello via a rectangle wave oscillator.

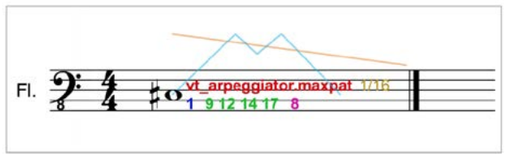


Figure 2. The first measure of the flute part in the meta-score. The low C♯ (fundamental) and its 9th, 12th, 14th and 17th stretched harmonics are rendered via the arpeggiator (1st instance). The arpeggio loop has 8 notes (each lasting 1/16th) and follows the 'mountain'-like profile shown in light blue. The amplitude envelope is the descending orange line.

In our line of work, the meta-score, that is, all the notes with all their parameters, is completely written 'by hand' in a *bach.score* object. [8] All the slot values are kept visible for each note; tied notes by default share the same slot value and are output as a single longer note. The first measure of the flute part is displayed in Fig. 2, with all metadata visible next to the notehead, as text or breakpoint functions. Depending on our desired operating mode, at any time, we can playback or render symbolically in real-time any portion of score. We can also render the whole meta-score off-line, in order to have it all output form *cage.meta.engine*'s right outlet. A screenshot of the main patch is displayed in Fig. 3.

Although our meta-score might appear bizarre at first (all instruments are notated in F clef, transposed one octave lower), it turns out to be extremely pertinent. For one thing, it is immediately clear that all instruments are somehow playing 'meta-unisons' (except for the right hand of the piano), correspondingly to the fact that all instruments are confined within the same harmonic series. When, at number 10, all fundamentals switch to the G, such important change is immediately readable in our meta-score, while it would take a larger effort of analysis to grab the same information from the score proper. Our meta-score also reveals very clearly the diminished seventh tetrachord (C♯, E, G, B♭) [9] underlying the whole construction, and abstracts the complexity of the arpeggi to a profile shape and a few indices (leaving most of the score made of long tied sequences of 4/4 notes).

5. CONCLUSIONS

Meta-score rewriting operations are a valid analysis tools, allowing significant properties of the musical content to emerge (in our case, harmonic relations and profiles). The fact that each instance of a process is symbolized by a note (or a chord) lets us represent pitch-based operations in a very intuitive way, via their 'most meaningful' pitch (or set of pitches), and shows the pertinence of representing a whole musical process through a single note.

On the other hand, meta-scores have a clear value in musical writing: not only do higher properties emerge, but they can also be easily handled rapidly and efficiently. One can, for instance, change the duration unit of all the notes in the arpeggi from 1/16 to 1/32, or transpose all the clarinet notes down a tritone with a simple Max message (see the *'modify?'* subpatch), or apply standard computer-aided composition techniques on higher-level structures (for instance, it would be fairly easy in our case to retrograde or stretch all the arpeggio profiles). All these possibilities make *cage.meta* a powerful tool for prototyping, exploring and composing.

Acknowledgments

The *cage* library has been developed by two of the authors within the center of electroacoustic music of the Haute École de Musique in Geneva, supported by the music and arts domain of the scene of the Haute École Specialisée of Western Switzerland.

6. REFERENCES

[1] G. Grisey, *Vortex Temporum*. Milan, Ricordi, 1995.

[2] J. L. Hervé, *Dans le vertige de la durée : Vortex Temporum de Gérard Grisey*. L'Harmattan, L'Itineraire, 2001.

[3] A. Agostini, E. Daubresse, and D. Ghisi, "*cage*: a High-Level Library for Real-Time Computer-Aided Composition," in *Proceedings of the International Computer Music Conference*, Athens, Greece, 2014.

[4] A. Agostini and D. Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal*, vol. 39, no. 2, pp. 11–27, 2015/10/03 2015. [Online]. Available: http://dx.doi.org/10.1162/COMJ_a_00296

[5] P. Nauert, "Timespan hierarchies and posttonal pitch structure: A composer's strategies," *Perspectives of New Music*, vol. 1, no. 43, pp. 34–53, 2005.

[6] M. Rohrmeier, W. Zudeima, G. A. Wiggins, and C. Scharf, "Principles of structure building in music, language and animal song," *Philosophical transactions of the Royal Society B: Biological sciences. CCCLXX:1664 (March 2015): Biology, cognition and origins of musicality*, vol. CCCLXX:1664, 2015.

[7] C. Agon and G. Assayag, "Programmation Visuelle et Editeurs Musicaux pour la Composition Assiste par Ordinateur," in *Proceedings of the 14th IHM Conference*. Poitiers, France: ACM Computer Press.

[8] L. Goehr, *The Imaginary Museum of Musical Works, An Essay in the Philosophy of Music*. Oxford, Claredon Press, 1992.

[9] N. Goodman, *Languages of Art. An Approach to a Theory of Symbols*. Indianapolis/Cambridge, Hackett Publishing Company, Inc., 1976.

[10] P. Schaeffer, *Traité des objets musicaux, Editions du Seuil*. Paris, 1966.

[11] C. Seeger, "Prescriptive and Descriptive Music-Writing," no. 44 (2), pp. 184–195, 1948.

[12] C. Nash, "The cognitive dimensions of music notations," in *Proceedings of the First International Conference on Technologies for Music Notation and Representation*, Paris, France, 2015.

[13] S. Emmerson, "Combining the acoustic and the digital: music for instruments and computers or prerecorded sound," in *The Oxford Handbook of Computer Music*, roger t. dean ed. New York: Oxford University Press, 2009, pp. 167–190.

[14] A. Sorensen and H. Gardner, "Programming with Time. Cyber-physical programming with Impromptu," in *Proceedings of OOPSLA10 : ACM International Conference on Object Oriented Programming Systems Languages and Applications*. New York: ACM, 2010, pp. 822–834.

[15] M. Mathews, "The Digital Computer as a Musical Instrument," no. 3591, pp. 553–557, 1963.

[16] M. Mathews, F. R. Moore, and J.-C. Risset, "Computers and Future Music," *Science*, vol. 183, no. 4122, 1974.

[7] On occasion, the flute part contains 12-TET tempered notes instead of 24-TET tempered notes. This is the case at measure 19, for instance: the natural C 'should' be a quarter-sharp C according to the harmonic series rendering. Grisey probably adjusted these notes to ease and respect the instrumental technique, but we did not account for these 'manual' adjustments in our work.

[8] By manipulating the schemes given in [2] one might build the meta-score content itself with *bach* via standard algorithmic techniques — this goes, however, beyond the purpose of this paper, which is to exemplify the usage of *cage.meta* via a specific case study.

[9] It should be pointed out that the E does not appear in the portion of score upon which we focused.

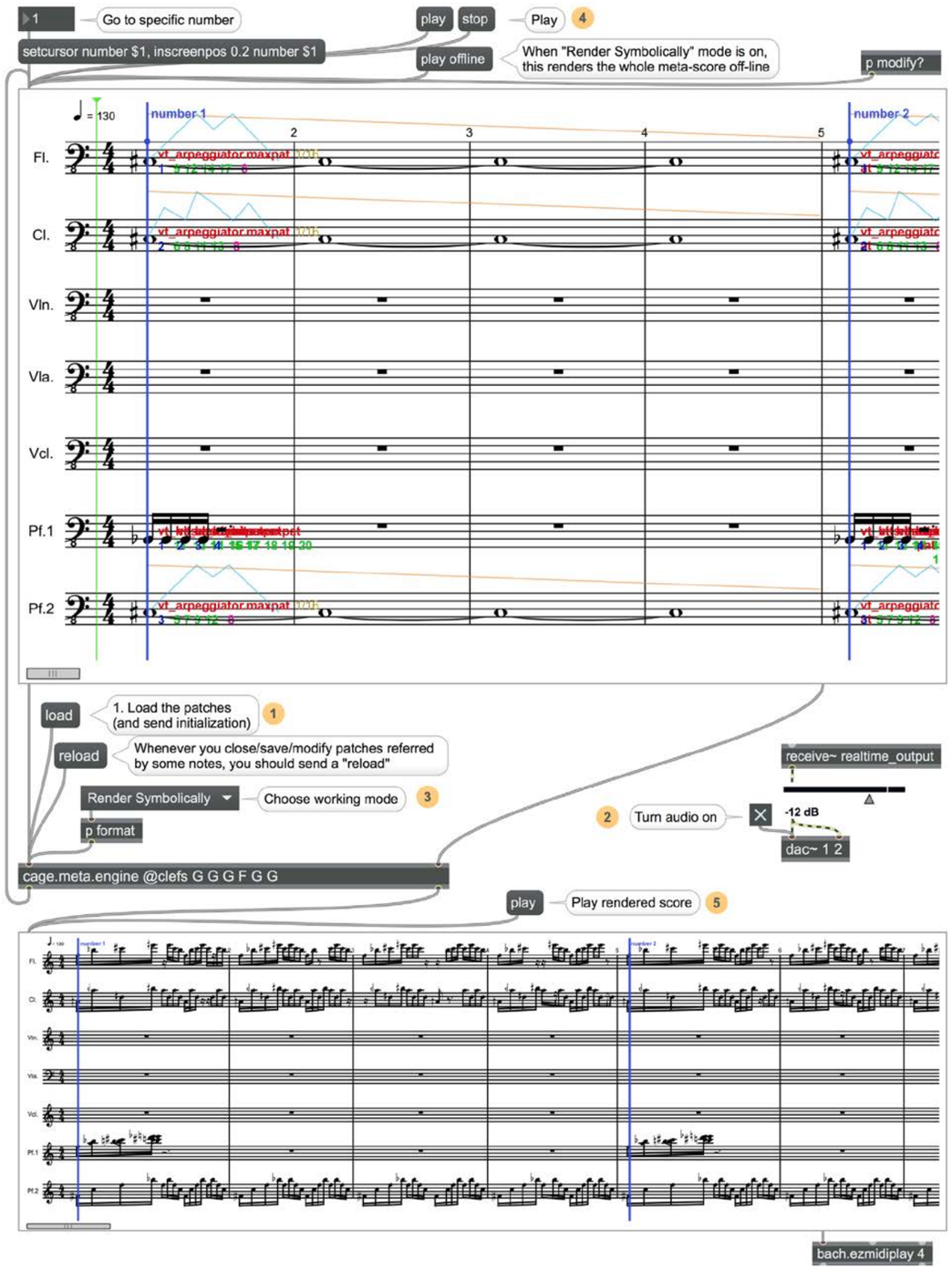


Figure 3. A screenshot of the main patch, showing the first measures of the meta-score modeling *Vortex Temporum*.

Composing in Bohlen–Pierce and Carlos Alpha scales for solo clarinet

Todd Harrop
Hochschule für Musik und Theater Hamburg
tharrop5@gmail.com

Nora-Louise Müller
Hochschule für Musik und Theater Hamburg
welcome@noralouisemuller.de

ABSTRACT

In 2012 we collaborated on a solo work for Bohlen–Pierce (BP) clarinet in both the BP scale and Carlos alpha scale. Neither has a 1200 cent octave, however they share an interval of 1170 cents which we attempted to use as a substitute for motivic transposition. Some computer code assisted us during the creation period in managing up to five staves for one line of music: sounding pitch, MIDI keyboard notation for the composer in both BP and alpha, and a clarinet fingering notation for the performer in both BP and alpha. Although there are programs today that can interactively handle microtonal notation, e.g., MaxScore and the Bach library for Max/MSP, we show how a computer can assist composers in navigating poly-microtonal scales or, for advanced composer-theorists, to interpret equal-tempered scales as just intonation frequency ratios situated in a harmonic lattice. This project was unorthodox for the following reasons: playing two microtonal scales on one clarinet, appropriating a quasi-octave as interval of equivalency, and composing with non-octave scales.

1. INTRODUCTION

When we noticed that two microtonal, non-octave scales shared the same interval of 1170 cents, about a 1/6th-tone shy of an octave, we decided to collaborate on a musical work for an acoustic instrument able to play both scales. *Bird of Janus,* for solo Bohlen–Pierce soprano clarinet, was composed in 2012 during a residency at the Banff Centre for the Arts, Canada. Through the use of alternate fingerings a convincing Carlos alpha scale was playable on this same clarinet. In order to address melodic, harmonic and notational challenges various simple utilities were coded in `Max/MSP` and `Matlab` to assist in the pre-compositional work.

We were already experienced with BP tuning and repertoire, especially after participating in the first Bohlen–Pierce symposium (Boston 2010) where twenty lectures and forty compositions were presented. For this collaboration we posed a few new questions and attempted to answer or at least address them through artistic research, i.e. by the creation and explanation of an original composition. We wanted to test (1) if an interval short of an octave by about a 1/6th-tone could be a substitute, (2) if the Carlos alpha scale could act as a kind of 1/4-tone scale to the BP scale, (3) if alpha could be performed on a BP clarinet, and (4) how would the composer and performer handle its notation? Some of these problems were tackled by computer-assisted composition. The final piece grew out of a sketch which was initially composed algorithmically, described in section 3.2.

Figure 1. Bohlen–Pierce soprano clarinet by Stephen Fox (Toronto, 2011). Owned by Nora-Louise Müller, Germany. Photo by Detlev Müller, 2016. Detail of custom keywork.

Our composition was premiered in Toronto and performed in Montreal, Hamburg, Berlin and recently at the funeral of Heinz Bohlen, one of the BP scale's progenitors; hence we believe the music is artistically successful considering its unusual demands. The quasi-octave is noticeably short but with careful handling it can be convincing in melodic contexts. Since the scales have radically different just intonation interpretations they do not comfortably coalesce in a harmonic framework. This is an area worth further investigation. Our paper concludes with a short list of other poly-tonal compositions albeit in BP and conventional tunings.

2. INSTRUMENT AND SCALES

This project would not have been possible with either a B♭ clarinet or a quarter-tone clarinet since the two scales described in 2.2 have few notes in common with 12 or 24 divisions of the octave. Additionally, our BP clarinet was customized and is able to play pitches which other BP clarinets cannot.

2.1 BP Clarinet

Our instrument is a unique version of a rare contemporary instrument, a Bohlen–Pierce soprano clarinet (see fig. 1) with two bespoke keys requested by the owner after a discussion with a colleague in Montreal. These are not specific to producing the Carlos alpha scale, but rather complement the existing BP scale notes with microtonal inflections. The BP clarinets are produced by Stephen Fox following a suggestion made in 2003 by Georg Hajdu [1]. At least ten regular BP soprano clarinets are owned by performers in Canada, the U.S.A. and Germany and we believe our model is unique among them due to its custom keywork.

2.2 Scales

Following are brief descriptions of the equal-tempered (e.t.) and just intonation (j.i.) varieties of the BP and alpha scales. [1] We shall express intervals as fractions, e.g., a 5/3 major sixth, chords as a set of frequency ratios, e.g., a 4:5:6 major triad, scales in a shorthand such as 12ed2 for 12 equal divisions of the 2/1 octave, and use 'c' for Ellis's *cents* value. [2] Furthermore we will look at chromatic-like sets of each scale rather than diatonic-like subsets.

2.2.1 Bohlen–Pierce

By way of combination tones, continued fractions and intonational experiments the BP scale was independently discovered between the early 1970s and early 1980s by Heinz Bohlen [2], Kees van Prooijen [3, pp. 50–51] and John Pierce et al. [4].

Rather than approximate the 4:5:6 major triad with the 0th, 4th and 7th steps from 12 equal divisions of the 2/1 octave, the e.t. BP scale approximates a 3:5:7 'wide triad' with the 0th, 6th and 10th steps from 13 divisions of a 3/1 *tritave*. In other words the core BP triad corresponds with the 3rd, 5th and 7th partials of a harmonic series and sounds like a stable if not consonant combination of pure major sixth and flat minor tenth above a root. Convention dictates that inversions occur at the twelfth rather than the octave, e.g., the first inversion of 3:5:7 is 5:7:9. The e.t. BP step size (we shall avoid the term 'semitone') is almost a 3/4-tone (see eq. 1), and since it is more than 100c the BP scale may thus be called a *macro*-tonal scale.

$$\frac{1200c}{\log 2} \cdot \log\left(\frac{3}{1}\right)^{\frac{1}{13}} \approx 146.3c \qquad (1)$$

Since all of its j.i. intervals are expressed as ratios using combinations of the primes 3, 5 and 7 but not 2, BP is neither a proper 7-limit scale nor is it able to express the familiar intervals of the 3/2 fifth, 4/3 fourth, 5/4 and 6/5 thirds, 8/5 minor sixth, 9/8 and 16/15 seconds, and 15/8 major seventh. On the other hand it expresses astonishingly well other intervals whose simple frequency ratios are composed of odd integers only, e.g., the 9/5 minor seventh and various *septimal* intervals such as the 9/7 major third, 7/3 minor tenth, 7/5 natural tritone and 15/7 minor ninth. Coïncidentally two of BP's j.i. intervals sound like our customary e.t. minor third and major tenth: the 25/21 BP second ≈ 301.8c, and 63/25 BP eleventh ≈ 1600.1c. BP's corresponding e.t. intervals, however, are slightly further away at 292.6c and 1609.3c, respectively.

[1] Although we will not discuss the tempering out of commas we will use the term e.t. as equal-division of a reference interval, or as equally-sized step size.

[2] As a frequency ratio, $1c = \sqrt[1200]{2}$ giving 1200c per octave and 100c per semitone.

2.2.2 Carlos Alpha

In contrast to the BP scale the alpha scale was designed to play the conventional 4:5:6 major triad better than in 12ed2—but at the sacrifice of the octave. It was discovered between the late 1970s and mid-1980s, also independently, by Prooijen [3, p. 51] and Wendy Carlos [5]. Equation 2 shows the common, simplest definition of alpha as 9 equal divisions of a perfect fifth though Carlos initially looked at dividing a minor third in half and then in quarters. Both authors give a rounded figure of 78.0c as step size and Benson suggests bringing the thirds more in tune by slightly tempering the fifth and using a step size of 77.965c [6, p. 222].

$$\frac{1200c}{\log 2} \cdot \log\left(\frac{3}{2}\right)^{\frac{1}{9}} \approx 77.995c \qquad (2)$$

There is no standard j.i. version of the scale beyond Carlos's set of target intervals: the 5/4 and 6/5 major and minor thirds, 3/2 perfect fifth, and 11/8 natural eleventh. Her target interval of a 7/4 natural seventh is not achievable in alpha, however an octave-inverted 8/7 septimal major second can be found instead. This accounts for five out of nine intervals within a perfect fifth. For our research we used computation to multi-dimensionally search for candidate j.i. intervals to fill in the harmonic space of the alpha scale and hopefully form bridges between alpha and BP.

3. COMPOSITION

3.1 Scalar Representation

3.1.1 Staff

The BP scale is always given as thirteen notes spanning a tritave, however, figure 2 shows only the first nine notes of the BP scale up to our experimental interval of equivalency of 1170c. Underneath is an alpha scale beginning on the same 'tonic' of F4 and also ending at 1170c on its sixteenth note. It would appear that the first and last notes are a major seventh apart but this is a shortcoming of our conventional notation system when working with microtonality, especially with non-octave scales. Interpreting the cents' deviations above the pitches informs us that F4 is about a 1/5th- or 1/6th-tone sharp and E5 is a 1/6th- or 1/5th-tone flat, spanning a range much closer to an octave than a major seventh.

Although the figure shows specific sounding pitches this notation was impractical for either composer or performer for these reasons: (1) intervals, let alone music, could not be easily read or written, (2) there *appeared* to be little consistency between pitches, and (3) the performer preferred another notation based on natural and alternative fingering.

We also discovered that our BP and alpha tonics differed by 7 or 6 cents. This was due to the clarinet using two reference tones on which to build each scale: the BP scale extended from 442 or 443 Hz (A4), whereas the alpha scale extended from 261.63 Hz (C4). Although D♯5 differs by

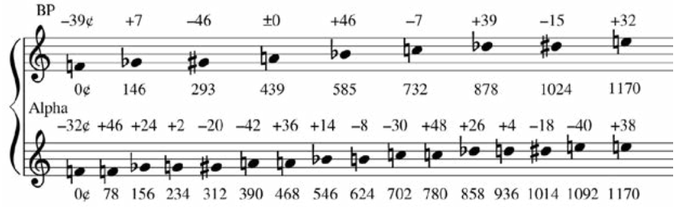


Figure 2. Bohlen–Pierce (top) and Carlos alpha (bottom) scales encompassing nearly one octave. Numerals indicate deviation from standard pitch (top) and interval from 'tonic' (bottom), in cents.

only 3c between scales, E5 was chosen as tonic (or F4 by quasi-octave transposition) as this allowed for neighbouring notes to be better matched and function as pivot tones between the two scales.

3.1.2 Quarter-tone–like Pivots

Despite the discrepancy in the tonics they and their neighbours were used as melodic pivot tones for modulating between scales. The first BP notes above and below the tonic were close enough to the second alpha notes above and below, thereby giving three common pitches at each tonic. More would have been welcome but beyond these the scales diverged and re-converged at the next tonic 1170c away. Therefore the usefulness of alpha as a quarter-tone–like scale to BP was limited. [3]

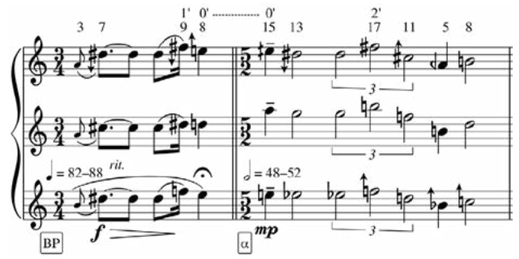


Figure 3. Score, mm. 20–21. Three representations of same music: sounding (top), MIDI (middle), and fingering (bottom); in Bohlen–Pierce (m. 20) and Carlos alpha (m. 21) scales.

Figure 3 shows the first modulation in the score from BP to alpha using pivot tones. The top staff shows sounding pitches and has two rows of arabic numerals for scale degrees. Both rows are the same notes but since this passage happens to hover around the ends of the scales, the tonic is shown both as *0'* or *8* for BP and *0'* or *15* for alpha. BP's *7, 9* and *8* correspond with alpha's *13, 17* and *15*. Slight discrepancies can be seen between the E5's and F♯5's due to the differing reference tones on which the BP and alpha scales were based, as mentioned earlier.

3.2 Integer Representation

As seen at the top of figure 3 simple integers proved practical for composing motifs and chords on paper—without musical staves—by assigning a number to each scale note. This was also useful in an early sketch which was algorithmically composed. The two sets of scale notes were interwoven in a 120 note super-scale (of 9.75 cent steps) with two modes: every 15th step made a BP scale, and every 8th step made an alpha scale. Harmonic movement was achieved by cross-fading the *probabilities* of chord notes being played rather than their volumes or dynamics. This often resulted, however, in some mingling of BP and alpha whenever a cross-fade was not instantaneous. Therefore the method was abandoned in order to keep the two tunings separate for the benefit of the listeners and performer.

[3] We use the term 'quarter-tone–like' to imply a scale whose notes fit exactly in between another scale's, like 24ed2 to 12ed2. If BP is 13ed3 then 26ed3 would be the more accurate choice. Incidentally there may be more prospect in 39ed3 [7, "Erlich's Triple BP Scale"] or 65ed3 [8], i.e. splitting BP into third- or fifth-tones.

Figure 4. Various BP calculators to convert between sounding, MIDI and fingering notations of a given pitch.

Eventually music had to be written so a series of calculators in `Max/MSP` assisted the composer in transcribing

from one notation style to another (see fig. 4). Sometimes up to five staves were necessary, each with a different notation style. Grouped into four types they are: (4a) desired pitch, (4b) sounding pitch notated in 1/8th-tones for convenience, (4c) MIDI keyboard pitches for compositional work, and (4d) so-called 'fingering pitches' easiest for a clarinet player to read and execute (e.g., see fig. 3, bottom staff). Only one staff sounded as it looked and the inclusion of four others, each with its own notation style, made the compositional process laborious.

3.3 Harmonic Representation

After the composition was completed we were still curious as to how to represent the union of two microtonal scales in harmonic space, beyond identifying affinities between superficial melodic pivot points, in case of future work with these or other combinations of scales.

3.3.1 BP Lattice

The j.i. BP scale is customarily shown on a square lattice with its 13 chromatic pitches delineated within a symmetrical diamond. This arrangement can be tiled to show groups of 'extended' BP pitches or enharmonics which are higher or lower than the reference pitches by various multiples of 7 cents. This lattice is a 3-d space projected onto a 2-d plane with axes representing primes 5 and 7, [4] and the axis for prime 3 flattened because the interval 3/1, the *tritave,* is the interval of equivalency in BP theory. [5]

It would be possible then to show the extended-BP pitch closest to any given alpha pitch, however, we found that this representation did not treat each scale fairly, not when most alpha pitches were plotted far from the tonic origin without contiguous intervallic connections to the BP core; nor was this model appropriate for our artificial scale since the standard BP set includes five more ratios, between the octave and tritave, which we had rejected when we limited our experiment to the interval of 1170 cents. Therefore the extended BP lattice was not used.

3.3.2 11-limit Lattice

In her alpha, beta and gamma scales Carlos sought to express ratios involving primes 2 through 11—not just 3, 5 and 7—therefore a 4-dimensional lattice with prime 2 flattened seemed more appropriate to us in depicting alpha and BP ratios together within a single model.

A `Matlab` program was made to calculate the thousands of ratios located as points in a hypercube and reject those that did not meet the following criteria: the ratio needs to be (1) within one quasi-octave of about 1170c, (2) within a tolerance range of error from the nearest e.t. scale step, and (3) factorially simple according to Tenney's harmonic distance metric [9] given in equation 3.

$$HD(f_a, f_b) \propto \log(a) + \log(b) = \log(ab) \qquad (3)$$

Setting the range was simple, allowing for a little stretch of a few cents' tolerance at either end. Setting the tolerance and harmonic distance involved more tweaking and, in the end, separate settings were used for BP and alpha otherwise far too many candidate intervals would be found to depict in the lattice (see fig. 5). In other words there can be more than one ratio appropriate for most steps of alpha. And although j.i. BP has a reference set of standard ratios some of its lesser-known enharmonics are actually closer to corresponding e.t. scale steps. E.g., the standard 27/25 is 13c away from the e.t. BP first, but the 49/45 alternative is only 1c away [10, p. 190]. [6] The same holds true for their inversions (at the tritave of course).

We wanted harmonic compactness, i.e. not having ratios floating untethered in space despite their sounding closer to an e.t. interval. Often these ratios were quite complex therefore there was a trade-off between simplicity and accuracy. Tenney's harmonic distance function (see eq. 3) was one of the criteria for filtering candidate ratios. [7]

For the Bohlen–Pierce set of ratios a tolerance of 9.35 cents was required for the BP second (25/21) and a harmonic distance of 11.85 was needed to catch the complex BP fifth (75/49). This yielded ten pitches including the complex 49/27 as an enharmonic alternative to 9/5.

For the alpha set the allowable prime numbers expanded from [3,5,7] to [2,3,5,7,11]. The tolerance was set much lower, to 6.6 cents, and the harmonic distance was set to either 10.59 to catch an accurate 55/28 quasi-octave or simply 10.26 to coïncide with BP's standard 49/25. Other enharmonic pairs were the first and second intervals of alpha as can be found in the chart.

Compared to typical lattices this structure in black, which represents our particular interpretation of Carlos alpha in j.i., appeared somewhat sparse and pokey. Unfortunately the BP scale, whose ratios are depicted in red, did not satisfyingly coalesce in this model.

A more sophisticated presentation could probably be put together using our quasi-octaves as *unison vectors* after gleaning articles by Fokker and Erlich. The interested reader is directed to Erlich's "Partch's 43-tone scale as a periodicity block", 1999, *Onelist Tuning Digest* 463–65; and to study Joe Monzo's "Lattice Diagram of 11-Limit Tonality Diamond", 1998; both at http://tonalsoft.com.

We attempted to show in figure 5 lines in red corresponding to BP's 5/3 and 7/5 axes in contrast to alpha's axes in black, which are 3/2, 5/4, 7/4 and 11/8. The problem was that in our model the vectors representing 5/3 and 7/5 were not in a ninety degree angle to each other, or any angle acute enough to easily distinguish them. We could also imagine lines connecting BP ratios *B5–6–7, 0–1, 3–4* and *7–8* but these are related by the ratio 49/45, not one of our primary ratios described above therefore no lines were drawn in the figure.

Instead we show dashed lines which connected the alpha and BP lattices and were interesting as potential harmonic bridges between them. Black lines link BP and alpha, red lines link BP and alpha *or* BP.

[4] The axes are actually 7/5 and 5/3 for the diamond configuration.[7]

[5] For comparison the regular chromatic scale can also be depicted on a lattice with axes for primes 3 and 5, and the axis for prime 2 flattened because 2/1 is the octave and taken for granted as interval of equivalency.

[6] Krantz and Douthett use Bohlen's *hekts* instead of cents, where 1 hekt equals 1/100th of a BP step.

[7] For comparison with the conventional 12-note chromatic scale we would set a Tenney harmonic distance of 10.5 in order to catch a tritone (45/32) and a rather high tolerance of 15.7 cents to capture the traditional thirds and sixths. These limits would also catch two enharmonics for the minor third and major sixth: 32/27 as alternative for 6/5, and 27/16 for 5/3. Although these ratios are more complex they are also ten cents closer to e.t. scale steps of 300 and 900 cents.

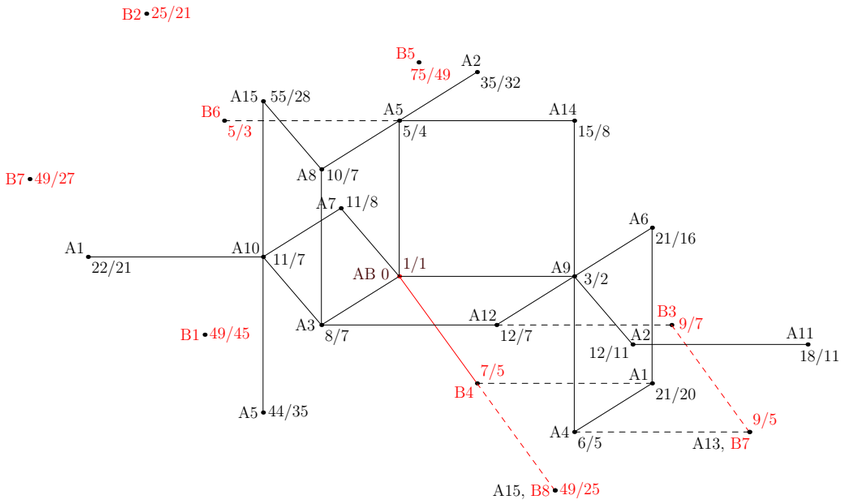


Figure 5. Lattice showing 11-limit ratios, attempting to combine alpha (black 'A's) with BP (red 'B's).

An advantage to visualizing ratios on a lattice is to see geometric patterns which correspond with intervals or entire chords in j.i. When scale steps are uneven it is not obvious which sets of intervals will sound the same. In e.t. scales of course all steps and combinations of steps are even. Our j.i. alpha scale had a septimal second of 8/7 between *A0–3, 1–4, 2–5,* not *3–6* nor *4–7,* but *5–8, 6–9, 7–10* etc. It was also easily apparent that there were two identical tetrads when recognized as parallelograms: (0358) and (1469), whose ratio sequence is 28:32:35:40, the obvious subset of which is a 7:8:10 'wide second–diminished fifth' triad.

3.4 Rhythmic Tiling Canon

A chance encounter with the mathematician Herbert Spohn during our residency encouraged us to include a rhythmic tiling canon which occurs near the end of the composition. It is a three-voice passage for one acoustic clarinet built on a three-note motif, and each voice is related by augmentation corresponding to the rhythm of the motif such that no simultaneous onsets of pitches would occur nor any overlap of notes. Thereby the clarinettist can perform the passage without multiphonics, singing, delay effects or pre-recorded parts. Although this paper is focused on poly-microtonal aspects we mention this canon as a relevant adjunct within the framework of computer-assisted composition.

4. DISCUSSION

We are satisfied with *Bird of Janus* as a musical composition however it can be seen that a harmonic representation of non-octave scales, which might potentially show patterns not otherwise apparent, is not easily achieved. Neither the Bohlen–Pierce nor the Carlos alpha scale has a 2/1 octave and the closest interval of 1170 cents can be interpreted as 49/25 as it is in BP or as 63/32 or even 55/28 as we considered when graphing alpha. Perhaps 63/32 would be best since the denominator is a power of two, suggesting a fundamental nearly six octaves below.

Furthermore since BP theory states that the 3/1 tritave replaces the 2/1 octave as interval of equivalency, but our experiment stops short of the octave, it is difficult to consolidate both scales into a cohesive harmonic lattice. Were we to experiment further in combining BP with another scale we might consider cheating with any interval that was perceptually close enough to a BP interval, i.e. including BP ratio doppelgängers from an 11-limit j.i. system. Using our limits described earlier this would yield alternatives listed in table 1. These are less than 5 cents out from equal-tempered BP, would probably sound the same to any listener, and be better connected in our 4-dimensional harmonic lattice. In fact the 11-limit alternative for a 49/45 BP first can already be found in figure 5 as a 12/11 alpha second.

Interval	BP	Alternate
1	49/45	12/11
2	25/21	32/27
5	75/49	32/21

Table 1. Sample of alternate ratios within 5c of e.t. BP steps.

Both BP and alpha are modern scales with atypical challenges for composers and performers. There are several works for alpha using synthesizer, notably by Carlo Serafini, but we are not aware of any for acoustic instruments. There is a growing body of acoustic concert music in BP especially from Germany and North America, for clarinets and various other acoustic instruments but none combining BP with alpha. Instead there is a handful of works that combine BP with the standard scale, i.e. 13ed3 with 12ed2: *Night Hawks* by Fredrik Schwenk, *Pas de deux* by Sasha Lino Lemke, and *Re: Stinky Tofu* by Roger Feria. An appropriate lattice could be designed to accentuate harmonic structures bridging these two scales as a tool, e.g.,

for microtonal music theory analysis.

The lattice has been enormously helpful to artists such as Ben Johnston, Jim Tenney and Erv Wilson to name but a few. Given that microtonal notation of higher-limit harmony and subsets thereof is daunting for many composers we believe that a harmonic lattice and notation software could be of interest to those willing to give it a try. For scale analysis there already exist superb programs such as `Tonescape` by Joe Monzo, `L'il Miss' Scale Oven` by X.J. Scott and `Scala` by Manuel Op de Coul.

5. CONCLUSIONS

Musical staff notation remains a delicate issue for composers and performers however there is at least one promising solution for the Max/MSP environment: `MaxScore` by Georg Hajdu and Nick Didkovsky. It features the ability to change the notation style for each staff with a few clicks [11]. E.g., one line of music can be shown in 1/4- to 1/12th-tone resolution, as nearest j.i. ratios, in Extended Helmholtz–Ellis or Sagittal notation, in BP clarinet fingering notation, or even in a 6-line staff notation specifically designed for BP music [12]. The user may also customize his or her own notation style particular to a project. This may be the 'killer app' for allowing composers and performers each to read in their preferred notation style.

These tools will certainly be welcome when we create another work in mixed tunings, with or without octaves. Although the interval of 1170c is noticeably short of an octave we believe it is possible to stand in as interval of equivalency especially in a busy, melodic context. If two pitches this far apart are sounding simultaneously then the result will be quite dissonant however, as was done in *Bird of Janus,* motifs that repeat a quasi-octave away do not dwell on the false note but instead continue to move through the melody and dispel any discomfort the listener might have on those moments.

Of more practical concern was making sure that the alpha notes, played on an instrument not designed to play alpha, were consistent and stable enough in tone quality, and this often lifted our focus away from too much mathematical, intonational concern.

Finally, although Carlos alpha does not function as a 1/4-tone–like scale to BP the investigation was most welcome. Obviously 26ed3 would double BP's 13ed3 scale. Nevertheless we believe poly-microtonality to be a fertile area for new music creation, whether with one or both of the scales presented here or with others.

Acknowledgments

The authors acknowledge the support of the Conseil des arts et des lettres du Québec, the Canada Council for the Arts and the Goethe Institut during the collaboration period, and of the Claussen-Simon-Stiftung during the preparation of this paper.

6. REFERENCES

[1] S. Fox, "The Bohlen–Pierce clarinet project." [Online]. Available: http://www.sfoxclarinets.com/bpclar.html

[2] H. Bohlen, "13 Tonstufen in der Duodezime," *Acustica*, vol. 39, no. 2, pp. 76–86, 1978.

[3] K. v. Prooijen, "A Theory of Equal-Tempered Scales," *Interface*, vol. 7, pp. 45–56, 1978.

[4] M. V. Mathews, L. A. Roberts, and J. R. Pierce, "Four new scales based on nonsuccessive-integer-ratio chords," *Journal of the Acoustical Society of America*, vol. 75, p. S10, 1984.

[5] W. Carlos, "Tuning: At the Crossroads," *Computer Music Journal*, vol. 11, no. 1, pp. 29–43, 1987.

[6] D. Benson, *Music: A Mathematical Offering*. Cambridge University Press, 2008.

[7] H. Bohlen, "The Bohlen–Pierce Site: Web place of an alternative harmonic scale." [Online]. Available: http://www.huygens-fokker.org/bpsite/

[8] B. McLaren, "The Uses and Characteristics of Non-octave Scales," *Xenharmonikôn: An Informal Journal of Experimental Music*, vol. 14, pp. 12–22, 1993.

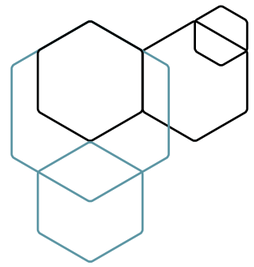
[9] J. Tenney, *Soundings 13: The Music of James Tenney*. Frog Peak, 1984, ch. "John Cage and the Theory of Harmony".

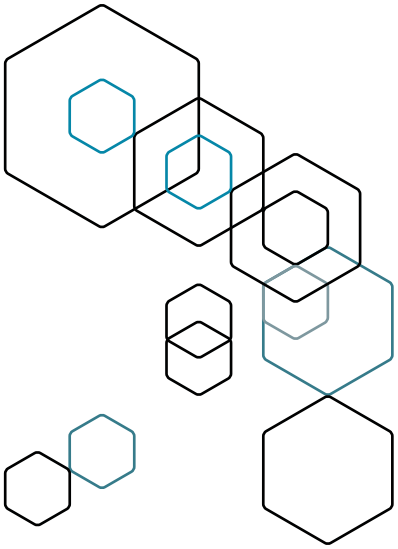
[10] R. Krantz and J. Douthett, "Algorithmic and computational approaches to pure-tone approximations of equal-tempered musical scales," *Journal of Mathematics and Music: Mathematical and Computational Approaches to Music Theory, Analysis, Composition and Performance*, vol. 5, no. 3, pp. 171–194, Dec 2011.

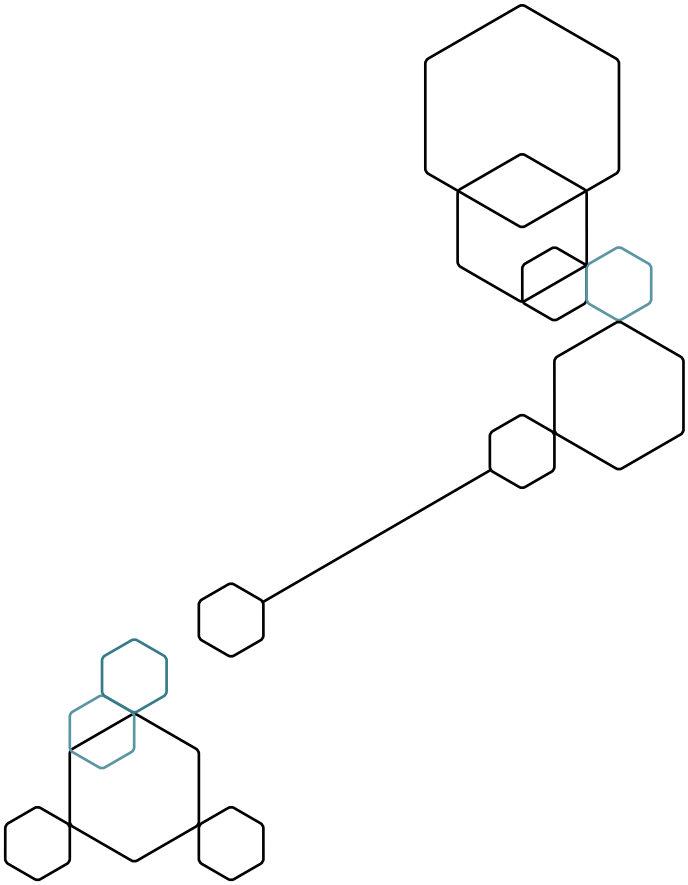
[11] G. Hajdu, "Dynamic notation - a solution to the conundrum of non-standard music practice," in *TENOR 2015: International Conference on Technologies for Music Notation & Representation*, Université Paris-Sorbonne and IRCAM. Paris: Institut de Recherche en Musicologie, IReMus, 2015, pp. 241–248.

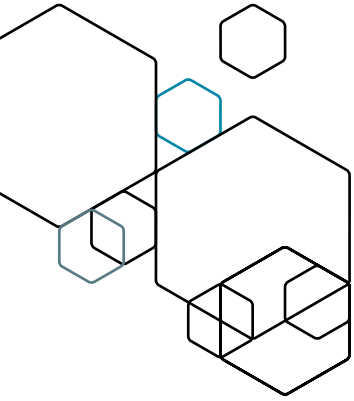
[12] N.-L. Müller, K. Orlandatou, and G. Hajdu, *1001 Mikrotöne 1001 Microtones*. Bockel, 2015, ch. "Starting Over – Chances Afforded by a New Scale", pp. 127–173.

Index Author A-Z









ICMC 2016

www.icmc2016.com