12º Simpósio Brasileiro de Computação Musical

12th Brazilian Symposium on Computer Music

SBCM

September 7-9, 2009
Recife - PE - Brazil

12°

Simpósio Brasileiro de Computação Musical

12th Brazilian Symposium on Computer Music

Proceedings of the 12th Brazilian Symposium on Computer Music

Anais do 12º Simpósio Brasileiro de Computação Musical

SBCM

NUCOM – Comissão Especial de Computação Musical da SBC
Sociedade Brasileira de Computação

http://compmus.ime.usp.br/sbcm/

7-9 September , 2009
Recife – PE – Brazil

Promoção / *Promotion*:

NUCOM/CECM Comissão Especial de Computação Musical da SBC

SBC Sociedade Brasileira de Computação

Organização / *Organization*:

USP

UFRGS UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

UFAC

VIRTUS IMPAVIDA UFPE

Suporte / *Sponsorship*:

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

CAPES

FACEPE Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco

RBV Rede Brasileira de Visualização

Apoio / *Institutional support*:

PORTODIGITAL

JDTL

EMF ELECTRONIC MUSIC FOUNDATION

ircam Centre Pompidou

D'ACCORD MUSIC SOFTWARE

GOL Linhas aéreas inteligentes

Simpósio Brasileiro de Computação Musical (12. : 2009 : Recife)
Proceedings of the 12th Brazilian Symposium on Computer Music, September 7-9, 2009, Recife, Brazil / eds. Marcelo Pimenta, Damián Keller, Regis Faria, Marcelo Queiroz, Geber Ramalho, Giordano Cabral ; org. tec. Marcelo Pimenta ; org. mus. Damián Keller ; org. local Geber Ramalho e Giordano Cabral ; org. geral Regis Faria e Marcelo Queiroz. – São Paulo : USP : SBC, 2009.
on-line + CD-ROM

ISSN 2175-6759
http://compmus.ime.usp.br/sbcm/

1. Computação musical (Congressos) 2. Computação aplicada (Congressos) 3. Música computacional (Congressos) I. Sociedade Brasileira de Computação II. Universidade de São Paulo III. Universidade Federal do Rio Grande do Sul IV. Universidade Federal do Acre V. Universidade Federal de Pernambuco VI. t.

CDD 004
786.76

Organization Committee

General Chairs

Regis Rossi A. Faria (LSI/Universidade de São Paulo and Organia Music Engineering)

Marcelo Queiroz (IME/Universidade de São Paulo)

Program Chairs

Technical Papers chair: Marcelo Pimenta (Universidade Federal do Rio Grande do Sul)

Musical Papers chair: Damián Keller (Universidade Federal do Acre)

Local Organization Chairs:

Geber Ramalho (Universidade Federal de Pernambuco)

Giordano Cabral (D'Accord Music Software)

Technical Papers Program Committee

Adolfo Maia Jr. (UNICAMP, Brazil)

Aluizio Arcela (UnB, Brazil)

Álvaro Barbosa (Universidade Católica Portuguesa, Portugal)

Andrew Horner (The Hong Kong University of Science & Technology, Hong Kong)

Chris Chafe (Stanford University, USA)

Eduardo Miranda (University of Plymouth, UK)

Emilios Cambouropoulous (Aristotle University of Thessaloniki, Greece)

Evandro Miletto (Universidade Federal do Rio Grande do Sul, Brazil)

Fabien Gouyon (INESC – Porto, Portugal)

Fabio Kon (Universidade de São Paulo, Brazil)

Flávio Soares Corrêa da Silva (Universidade de São Paulo, Brazil)

Geber Ramalho (Universidade Federal de Pernambuco, Brazil)

Gianpaolo Evangelista (Linköpings Universitet, Sweden)

Giordano Cabral (D'Accord Music Software)

Hugo de Paula (PUC-MG, Brazil)

Ian Whalley (University of Waikato, New Zealand)

Jean-Pierre Briot (CNRS - Université Paris 6, France)

Jonatas Manzolli (UNICAMP, Brazil)

Luis Jure (Universidad de la República, Uruguay)

Marcelo Soares Pimenta (Universidade Federal do Rio Grande do Sul, Brazil)

Marcelo Queiroz (Universidade de São Paulo, Brazil)

Marcelo Wanderley (McGill University, USA)

Marcio Brandao (UnB, Brazil)

Mikhail Malt (IRCAM, France)
Oscar Di Liscia (Universidad Nacional de Quilmes, Argentina)
Peter Beyls (Hogeschool Gent, Belgium)
Petri Toiviainen (University of Jyvaskyla, Finland)
Regis R. A. Faria (Universidade de São Paulo and Organia Music Engineering, Brazil)
Roger Dannenberg (Carnegie Mellon University, USA)
Rosa Viccari (Universidade Federal do Rio Grande do Sul, Brazil)
Sever Tipei (University of Illinois School of Music, USA)
Victor Lazzarini (National University of Ireland, Ireland)

Musical Papers Program Committee

Adina Izarra (Universidad Simón Bolivar, Venezuela)
Alejandro Viñao (United Kingdom)
Catalina Peralta Cáceres (Universidad de los Andes, Colombia)
Damián Keller (Universidade Federal do Acre, Brazil)
Daniel Teruggi (GRM, Institut National de l'Audiovisuel, France)
Fernando Iazzetta (Universidade de São Paulo, São Paulo, Brazil)
Fernando Lopez-Lezcano (CCRMA, Stanford University, USA)
Javier Alvarez (Conservatorio de las Rosas, Mexico)
Rodolfo Coelho de Souza (Universidade de São Paulo, Ribeirão Preto, Brazil)

Acknowledgements

Élyde Mauricio de Campos (Library of the Polytechnic School of University of São Paulo)
All volunteers and professionals involved in the local organization

Arts and creation

Totum Marketing e Comunicação

Proceedings Editorship

Regis R. A. Faria (digital version)
D'Accord Music Software (CDROM printing)

Letter from the Symposium Chairs

The beautiful city of Recife at the northeastern Brazilian cost welcomes again the Brazilian Symposium on Computer Music (SBCM). In these days when production and consumption of music has been transformed by the action of social networks and new paradigms of creation, distribution and consumption of music, new topics attract the interest of both academy and industry communities.

This 12th edition of the SBCM is dedicated to the relentless and continuous renewal in paradigms for making music with computers. A total of 46 original research works were submitted in three categories: 28 full technical papers, 11 full music papers, and 8 posters. Following the reviewing process and necessary adjustments, 10 technical papers, 5 music papers and 9 posters were accepted for publication, with an acceptance rate of 38% for technical papers.

A comprehensive technical and cultural programme was designed for all participants of SBCM 2009, including two concerts featuring musical pieces both invited and presented on the musical papers, and a discussion panel on the leading symposium theme, namely technologies and devices for making music, covering topics of impact for the computer music field in Brazil and abroad.

Aiming to encourage the participation of student works and innovative demonstrations of computer music works we proposed the On The Spot venue, open to all participants of the symposium. IRCAM consolidates its participation in SBCM by offering a tutorial on computer-aided composition, approaching formal models and software technologies for the creation of contemporary music. Additionally, we are grateful to IRCAM for offering a one-year subscription to IRCAM Forum for the best student paper.

We brought together speeches from experts of both art and technology worlds for a convergence of ideas in SBCM 2009. Special thanks go to our invited guest, professor Miller Puckette (University of California San Diego) and author of the graphical programming environment Pure Data (Pd). Prof. Puckette contributions to computer music have boosted in recent years a new generation of music creation habits featuring interactivity, real-time operation and intuitive interfaces.

We also give special thanks to the Colombian composer Catalina Peralta, another invited guest who honored us with an additional musical concert featuring remarkable works by Latin American composers.

Our results wouldn't be nearly as thorough if it weren't for our reviewers board's invaluable job. We had 48 reviewers from the Americas, Europe, Asia and New Zealand, completing a total of 168 reviews. We want to express our gratitude to the technical and music program committees, who have devoted precious time and delivered an excellent work, contributing to the scientific and technical excellence of SBCM publications.

We are also happy to announce the project of a future release from the Electronic Music Foundation on SBCM 2009 music works. Additionally, we are pleased to announce that a selection of the best papers presented in SBCM 2009 will be invited for a special edition of the Journal of New Music Research, to appear in 2010-11.

We hope these days at Recife will be unforgettable and wish all participants a pleasant stay.

Regis Rossi Faria, Marcelo Queiroz, Marcelo Pimenta, Damián Keller, Geber Ramalho and Giordano Cabral

Invited talks

New sounds from guitars

Miller Puckette

Now that it's easy to get multiple channels of sound into and out of a computer with pretty low throughput latency, an obvious and attractive application is signal-processing an electric guitar separately string by string. This permits a wide variety of non-linear processes which, if properly designed, preserve the periodicity of the individual string while making possible a wide variety of new sounds. Pitch-synchronous algorithms also become available. In many situations it is possible to separate the effect of amplitude change from that of changing harmonics, thus preserving the playability of the instrument. Several examples of this will be described here, including octave changes, wave bashing, formant generation, and neighboring-string intermodulation.

About Miller:

B.S. in Mathematics (1980) and Ph. D. in Mathematics from Harvard (1986). Former member of MIT's Media Lab from its inception until 1987, and researcher at IRCAM where he wrote the Max program for Macintosh computers, first distributed commercially by Opcode Systems in 1990. Joined the Music department of the University of California, San Diego in 1994, and is now Associate Director of the Center for Research in Computing and the Arts (CRCA). He is currently working on a new real-time software system for live musical and multimedia performances called Pure Data ("Pd"). Joined the Global Visual Music project in 1997 and since 2004 has performed with the Convolution Brothers.

Electroacoustic approach to some colombian composers

Catalina Peralta Cáceres

It will be a presentation about a selection of relatively recent electroacoustic and computer music pieces, by composers as Harold Vasquez, Roberto García, Mauricio Bejarano, Germán Toro, Catalina Peralta, Alexandra Cárdenas.

About Catalina:

Associate Professor at the Department of Music of the Faculty of Arts and Humanities of the Universidad de Los Andes in Bogotá, Colombia. Teacher of composition and electroacoustic music since 1996. She is Composer of the Vienna Academy of Music and Magister Artium at the University Mozarteum-Salzburg.

Tutorial I

Computer-Aided Composition - Formal Models and Software Technologies for Contemporary Music Creation

Jean Bresson, IRCAM (Institut de Recherche et Coordination Acoustique/Musique, Paris, France)

The research topics of the Music Representations group at IRCAM are centered around the formalization of musical structures and the conception of related computer models. Through many years of collaboration between scientists and musicians, the different projects of the group gradually constituted an original conception of the relations between music creation and computers. These projects resulted in computer-aided composition environments integrating computing and musical paradigms, used and shaped by a wide community of researchers and composers. Various general areas regularly come between these works, related for instance to the programming languages and paradigms, human-computer interaction, to representation and computation levels in musical structures, or to the modeling and representation of time in music.

As a general framework for computer-aided composition research and creation, it will be given an introductory overview of the OpenMusic computer-a ided composition environment. OpenMusic (OM) is a visual programming language dedicated to music composition. This environment allows composers to create programs graphically and to implement their own musical models and approaches. Some concrete example will be given of pieces written using OM to different extents.

The tutorial will then introduce more specific research directions and projects, including computational analysis and the study of algebraic structures in music, issues in the signal vs. symbolic musical representation and the control of sound synthesis, computer-aided orchestration, spatialization and symbolic interaction systems.

Through this presentation we try to emphasize a specific approach in music research but most of all a promising network of relations to other music technology areas such as digital signal processing, sound spatialization or real-time systems.

OpenMusic is designed and developed by G. Assayag, C. Agon, J. Bresson.

Tutorial II

Computer music meets video games
Giordano Cabral (MusiGames Studio / Daccord Music Software)

Although a considerable part of computer music systems, specially interactive musical systems, shares methods and problems with video game titles, both from a conceptual and technical point of view, their development models are clearly different.

The video game development cycle respects rules determined by an extremely strong industry, therefore specific tools and requirements appear. Game engines are widely spread, art quality is crucial, and there is almost an obsession for the gameplay. Interactive systems, on the other hand, usually follow an experimental model, looking for innovation instead of public acceptance.

However, these two worlds started to strongly interact since blockbusters appeared, like Guitar Hero, Rock Band, and Dance Dance Revolution. Computer music expertise became worthy on the video game market, but the video game tools and development models did not conversely influenced the computer music community.

This tutorial starts from MusiGames Studio experience in developing video game titles for multiple platforms (Nintendo Wii, Microsoft Xbox360, Apple iPhone, PC/Mac). The company is a brand name of Daccord Music Software, which has been developing music software for 9 years.

The tutorial will present some programming environments and tools to game creation; the reasons to use game engines, their advantages, drawbacks, and limitations; the creation of audio and signal processing specific libraries; and how the music information retrieval technology may contribute to create a new generation of musical games.

Workshop

Advanced Topics on Pure Data
Miller Puckette

On the Spot / Na hora

We are pleased to announce an open venue for participation in the SBCM: Na Hora / On the Spot. This venue will be available to artists, musicians and researchers who like to share their ideas in an informal, casual setting.

Participation is open to registered participants of the SBCM. There will be no screening process for these proposals, but time slots are limited.

The organizing committee will provide a basic setup (including PC, internet and sound system). Any special hardware or software requirements will have to be handled by the participants.

Panel / Round-table

New Paradigms for Computer Music

The new technologies, particularly mobile devices, Internet and Web have had a major impact in the production and distribution of music, defining new business models, enabling market niches, providing new forms of interaction among musicians and among them and the public. These technologies indeed open up new paradigms for computer music.

However, without losing the perspective of the future, we should recognize that some new paradigms are already being exploited and we have probably not paid enough attention to them. In fact, the easy access to technologies for music creation - coupled with the assimilation of research results in computer music - are revealing young artists, many of them grouped under the label "independent or alternative production". Unfortunately, this cultural and intellectual movement paradoxically does not seem to influence enough the academic research (in computing) or the creation of the so-called "electroacoustic music".

Is it possible to establish bridges between these communities - artists, computing research and music production - to improve cooperation? Could this cooperation leverage new paradigms in computer music?

Participants:

Geber Ramalho (UFPE) - moderator
Miller Puckette (UCSD)
Regis Rossi A. Faria (USP/Organia)
Marcelo Pimenta (UFRGS)
Damián Keller (UFAC)
Felipe Machado (CDTL)
"China" (artist)

Concerts

Concert I

Acercamiento electroacústico a algunos compositores Colombianos

Electroacoustic approach to some Colombian composers

Curator: Catalina Peralta

CATALINA PERALTA Bogotá-Colombia (1963)
She is Associate Professor at the Department of Music, Faculty of Arts and Humanities, of the Universidad de Los Andes in Bogotá, where she has teached composition and electroacoustic music since 1996. Composer of the Vienna Academy of Music. Magister Artium at the University Mozarteum-Salzburg. Participation in different Festivals with her works.

Program:

Rothko IV (Germán Toro, 13', 2008)
Electroacoustic composition. From the perspective that time has give us on his whole work, Rothko appears to me as an Artist who followed over the years a path towards abstraction to express the fullness of his subjects through space and color in a deep personal way, independent from daily ephemeral necessities of art business. In Rothko's work what is been shown acts likewise than what remains concealed. Form and color language rise from the reflection about his subjects: the Greek myths, the origin of tragedy, the structure of the psyche, the surrealism, the fresco paintings from Fra Angelico, etc. Those are not aesthetical end in itself. That, what after slowly distillation remains, contains the essence of its origin. Not only that, what is being seen is thus present but also that, what became outward dispensable. Rothko IV defines a clear syntax based on composition models that combine continuous and discontinuous elements. Discontinuous elements have the quality of recognizable sound objects that are combined in sequences leaving open spaces to perceive the sounds behind. Continuous sounds appear as surfaces and as a fluctuation processes. The piece is a further attempt to approach the idea of space as a superposition of layers that enter and leave the sound space covering and discovering further sound layers existing in the background and suggesting a process that continues beyond the limits of conscious perception.

Electra 2 (Mauricio Bejarano, 10'17", 2009)
"Voy a visitar a los pingüinos y les ofrezco un concierto de fonógrafo, ya lohabíamos hecho varias veces con éxito, pero esta vez el efecto sobrepasa todo lo imaginable.
… Y parecería que los pingüinos saben apreciar el talento … puesto que uno de ellos trató de meterse entre la bocina, probablemente para escuchar mejor."
(En Exploradores como nos gustan de Jean Charcot, citado por Julio Cortázar en Losautonautas de la cosmopista.)

Radioland (episodio 1) (Julián Jaramillo, 4'38", 2009)
After a year of researching and gathering material about radio in Colombia, this piece uses radio samples about politics, enterteinment, sports and humor, which tell little media stories. The piece is a trip through electromagnetic spaces from recent past, that reveals sonic imaginary tendencies and memories from the colombian construction of media.

FUXING (Juan Reyes, 13'14", 2006)
TapeMusic, originally in 4 channels (For Physical Model of a Banded Waveguide, Pipa and Gamelan). This is a computer generated media piece for Banded Waveguide physical models and spectral modeling of Pipa and Gamelan. Banded waveguides computer models are used for sound of instruments such as bowed marimbas and vibraphones. While searching for the timbres in this composition (and therefore the spectral modeling), the composer was aiming at sounds rich in odd harmonics and not necessarily tempered. Furthermore the rhythmic component was an issue, while looking for patterns which are only obtainable with

an expression machine algorithm. The sound sources in this piece make a thread of micro-tonal sonorities which also evolve horizontally as function of time. This composition was achieved with Juan Pampin's ATS and Common Lisp Music (CLM) on a PlanetCCRMA Linux environment.

Cristal verde muy oscuro, casi negro (Roberto García Piedrahita, 11'11", 2009)
Green glass very dark, almost black
Vitreous tool; vitreous ornament. Mirrors and arrows and spears.
Translucent. To polish. Obsession, siege, disturbance,
fixed idea; with tenacious persistence it assaults the mind.
Obstacle. Impediment, difficulty, disadvantage.
Lack of light to perceive the things. Place without light, or with
very little light. ...

Pieza electroacústica No 2 (Jorge Gregorio García M., 10'37", 2004)
The row material comprehends the recording and further spectral analysis of some structural chords played by a piano, produced in different ways, such as 'normally' played in the keyboard, on the strings, as resonance, etc. Some of the main harmonic components in the sounds, isolated, maintain their own spectral characteristics such as duration, intensity, place inside the harmonic complex, between others. These physical characteristic data of the sounds are interpreted as musical objects in itself in the structure of the work. As a matter of fact, these data will be modified and overlapped with other related material obtained by electronic means conserving some characteristics within each other (such as pitch, intensity, duration, etc.) working as structural bridges between the acoustic and the electronic data. In this way, the development of the natural harmonic spectrum of the original signals will behave as some sort of compositional maps, which would be interpreted in different compositional means, conserving and manipulating some data in order to create independent but related structures along the piece.

The composers:

Germán Toro: *born 1964 in Bogotá. Composition studies with Luis Torres Zuleta in Bogotá, Erich Urbanner and Karl-Heinz Füssl at the University of Music and Performing Arts, Vienna. Conducting courses with Karl Österreicher and Peter Eötvös. Studies on Electroacoustics and Computer Music in Vienna and at IRCAM in Paris. His catalogue includes instrumental, electroacoustic and mixed compositions. He has received composition grants from the Colombian Ministry of Culture, the Austrian Republic and the Experimental Studio Freiburg as well as composition prizes. His works have been performed in Europe, South Korea, North- und South America and in collaboration with Ensembles like die Reihe Vienna, On-Line Vienna, Mondrian Basel, and Klangforum Vienna among others. From 1999 to 2006 he was director of the computer music course at the University of Music and Performing Arts, Vienna, where he was guest Professor for Electroacoustic Composition 2006/07. Since October 2007 he is director of the ICST at the School of Arts, Zurich.*

Mauricio Bejarano: *Associate Professor at the Conservatory of Music of the National University in Bogotá, Colombia and Associate Professor in the faculty of the College Fine at the National University of Colombia. He has explored several fields of creativity such as design, sculpture, painting, poetry, essays and acousmatic art. He has attended several workshops and seminars on electroacoustic composition by Daniel Teruggi and François Bayle (Ina-GRM), Francis Dhomont, Stéphane Roy, John Chowning, Jean Claude Risset and Michel Zbar. He has exhibited his artwork both at individual and group exhibitions. His music is widely performed around the world including Spain, France, Holland, Belgium, Austria, Canada, USA, Mexico and Uruguay. His composition Jagua (r), was awarded a prize in the International Acousmatic Composition Competition, Noroit 95. Mauricio Bejarano has created radio phonic and sonic art works, has written several essays on electro acoustic music, and periodically lectures on the topic.*

Julián Jaramillo (www.julianjaramillo.net): *composer and intermedia artist. He has joined video, tv, radio, dance, opera, instalation, sound art and design, internet music projects. He is member of "El salón de la Justicia" team, he works from 2007 as a professor and researcher in Universidad Central. Julián lives and works in Bogotá.*

Juan Reyes: *composer and engineer has pursued degrees in Computer Science, Mathematics and Music aiming to Computer Music and related fields at the University of Tampa and at CCRMA in Stanford University. Among a variety of subjects in music and acoustics he has studied with John Chowning, Chris Chafe, Julius Smith III, Jonathan Berger, Brian Ferneyhough, Terry Mohn and Max Mathews. Recent interests steer to modeling of acoustical phenomena and musical expression plus information systems and human computer interface and their application to composition and performance. He has been professor of Music and Arts at La Universidad de Los Andes in Santafé de Bogotá, Colombia. At CCRMA his compositional work is focused toward methods and pieces for traditional acoustic instruments, live electronics, tape-media format and art-music sound installations.*

Roberto García Piedrahita: *born in Bogotá, Colombia in November 1958 and lived in Barcelona during the 80s. He studied composition and electroacoustic music with Chilean composer Gabriel Brncic at the Phonos Foundation in Barcelona. He graduated from the Universidad Nacional de Colombia in music composition. He is professor at the Conservatory of the Universidad Nacional de Colombia since 1993 where he has taught courses in Composition, Interpretation and Sound for Media; he also coordinates the Computer and Electroacoustic Music area. He is currently teacher in two Masters: Media and Theater and Interdisciplinary Arts at the Faculty of Arts.*

Jorge Gregorio García M.: *born in 1975, Bogotá, Colombia. He is Associate Professor at the Department of Music of the Faculty of Arts and Humanities of the Universidad de Los Andes in Bogotá. Composer and theoretician, graduated from Los Andes University in 2000. In 2003 he earned his Masters Degree in Music Composition and Theory at the Texas Christian University. His Pieces have been performed in different contemporary music festivals in North America, South America and Europe. Education: 2003 Composition/Analysis seminar/workshop with composer in residence Dr. Brian Ferneyhough. University of North Texas, Denton, USA. 1999 "Workshop Seminary of popular and academic music composition" conducted by Maestro Coriún Aharonián, Bogotá, Colombia. 1999 "Composition seminary/workshop of academic Latino American music analysis", conducted by Maestra Graciela Paraskevaídis, Bogotá, Colombia. Awards Graduate Assistantship, granted by the Graduate School of Arts at Texas Christian University, Fort Worth, TX, 2000-2002. Consecutive winner of the 2002 and 2003 Jonathan Durington Student Composition Awards, Fort Worth, Texas, USA.*

Concert II

SBCM 2009 concert
Curador / Curator: Damián Keller

Programa / Program:

Context (Miller Puckette, 5', 2009) (estréia)
Formato: *guitarra, processamento em tempo real*
Esta miniatura pertence a uma série de trabalhos cuja idéia é investigar o processamento sonoro das seis cordas da guitarra de forma individual - uma técnica que abre muitas possibilidades além dos efeitos "clássicos". Tradicionalmente os efeitos operam num único sinal pré-mixado. Em Context, um algoritmo simples de inteligência artificial controla - em tempo real - o som de cada nota em função do seu contexto melódico e textural.

10°29´N (Adina Izarra, 4', 2007)
Formato: *estéreo*
A peça 10°29´N é baseada em pássaros venezuelanos: Tordos, Guacharacas, Guacamayas. Há também a chamada muito característica do amolador, uma tradição antiga da Galícia espanhola que ainda está presente nas ruas de Caracas. O título faz referencia a posição geográfica da cidade, situada a 10º 29´ ao norte do equador.

(Un)folding (Daniel Barreiro, 12’ 20”, 2004)
Formato: *estéreo*
(Un)folding é baseada principalmente em sons de papel. A ação de dobrar papel foi uma maneira não apenas de produzir os sons, mas ofereceu também idéias para a organização do material musical da obra. A idéia é que, quando se dobra uma folha de papel, o verso da folha é revelado. Dobrando-a repetidas vezes, é possível obter várias formas que podem ser continuamente transformadas. Da mesma maneira, (Un)folding procura mostrar diferentes pontos de vista de um objeto que se move e se transforma durante o desdobrar do tempo. A peça foi Finalista no VI CIMESP, em 2005.

Freedom in Hot & Cold / Libertad en caliente y frío, V. 2. (Catalina Peralta, 4’ 32”, 2007)
Formato: *estéreo*
Parte das amostras utilizadas foram gravadas em uma das principais ruas de Bogotá, Carrera 7tima, na quinta-feira 5 de julho de 2007, durante uma manifestação espontânea. As pessoas - de diferentes classes e pensamento político - protestavam contra o assassinato de 11 deputados do Valle, sequestrados a 9 anos atrás pelas FARC. As pessoas protestavam contra as FARC, a violência, o sequestro, os conflitos armados, pedindo paz e a liberação de todos os reféns. As outras amostras foram retiradas do lado frio do mundo: sons quebrando o gelo (samples do Jens Hedman) e som de gotas e de água corrente (por Natasha Barrett).

Canudos (Liduino Pitombeira, 6’ 16”, 2000)
Formato: *estéreo*
Canudos foi uma guerra religiosa e civil que aconteceu no Brasil, no final do século 19. A peça é cheia de elementos místicos que representam a religiosidade de Antônio Conselheiro, retratada através do crescendo estrutural da obra. Os momentos finais mostram um episódio marcante em que o exército de dois mil homens luta contra cinco rebeldes. A peça usa phase vocoding, linear predictive coding, e síntese FM, subtrativa e granular. O software utilizado foi Csound, Peak e ProTools. Canudos foi realizada no Music & Art Studio Digital (MadStudio) da Louisiana State University.

Green Canopy: On the Road (Damián Keller, Ariadna Capasso, Patricia Tinajero, Luciano Vargas Flores, Marcelo Soares Pimenta, 7’, 2009) (estréia)
Formato: *versão estéreo de obra ubíqua*
Green Canopy é uma série de obras escultóricas e sonoras que envolve elementos extraídos da floresta ocidental amazônica. Green Canopy, On the Road é a obra mais recente dentro dessa série. On the Road fornece material sonoro para aplicação de um conceito embasado nas técnicas de interfaces humano-computador: a sonda musical. As sondas musicais são dispositivos que permitem estabelecer uma experiência pessoal para cada ouvinte da obra musical. Esta versão mostra uma das tantas possibilidades de realização da obra.

Kitchen <-> Miniature(s) (Fernando Lopez-Lezcano, 9’43”, 2005/2006)
Formato: *24 canais*
Um gravador de boa qualidade e uma cozinha. A humanidade sintonizada a formas e tamanhos comuns que criam ressonâncias compartilhadas: em toda parte, há uma cozinha. Estas miniaturas exploram alguns utensílios de cozinha e muitos dos pequenos aparelhos que gravei (ou seja, qualquer coisa que coubesse comigo dentro do armário do meu quarto). Tem destaque na peça: o contador mecânico de uma torradeira, embalagens de biscoito, as ressonâncias internas da tampa de uma frigideira, e muitos outros instrumentos de cozinha. Mais de 3000 linhas de código Common Lisp são usadas para criar grandes formas e detalhes no processamento do som. Agradecimentos a Bill Schottstaedt (Common Lisp Music), Juan Pampin (Analysis, Transformation and Synthesis) e Rick Taube (Common Music).

sonificacao~ 7 9 9 2009 (Pedro Patrício, 8' 30", 2009) (estréia)
Formato: *estéreo, processamento em tempo real*
[pedro patrício(
[sonificacao~ 7 9 9 2009]
[filmes para música $1(
[sbcm]

Os compositores / The composers:

Adina Izarra *(Caracas, Venezuela, 1959) obteve o seu PhD pela University of York no ano 1989. Em 2002 passou a formar parte do "Colegio de Compositores Latinoamericanos de Música de Arte". E atualmente leciona na Universidade Simón Bolivar, onde dirige o Laboratorio Digital de Música.*

Catalina Peralta *(Bogotá, Colômbia, 1963) é Professora Associada do Departamento de Música da Universidade de Los Andes, em Bogotá, onde ensina composição e música electroacústica desde 1996. É compositora formada pela Escola Superior de Música de Viena, e Mestre Artium pela Universidade Mozarteum de Salzburgo.*

Damián Keller *(Buenos Aires, Argentina, 1966) é coordenador do Núcleo Amazônico de Pesquisa Musical e professor adjunto na Universidade Federal do Acre. As suas áreas de interesse incluem a ecocomposição e a música ubíqua.*

Daniel Barreiro *(Brasil, 1974) nutre interesse especial pela música eletroacústica. Sua produção inclui obras acusmáticas e obras eletroacústicas mistas. É Doutor em Composição pela University of Birmingham (Inglaterra). Atualmente atua como Professor Adjunto na Universidade Federal de Uberlândia.*

Com Mestrado em Engenharia Eletrônica (Universidad de Buenos Aires, Argentina) e Música (Conservatório Nacional Carlos Lopez Buchardo), **Fernando Lopez-Lezcano** *tem trabalhado no campo da música eletroacústica desde 1976. Ele criou e administra o CCRMA Planet at Home, um pacote de aplicativos livres para música e som na plataforma Linux. A sua música encontra-se disponível em CDs publicados em América, Europa e no Sudeste Asiático.*

Apesar da sua formação como matemático (BS, MIT, 1980; PhD., Harvard, 1986), **Miller Puckette** *tornou-se pesquisador da área de computação musical. Ele trabalhou no MIT Media Lab desde a sua criação até 1986, e posteriormente no IRCAM (Paris, França) 1986-1994. Miller Puckette atualmente leciona no Departamento de Música da University of California, San Diego, onde também atua como Diretor-Adjunto do Center for Research in Computing and the Arts (crca.ucsd.edu). Em 2008, foi premiado com o Seamus Prize em reconhecimento à sua ampla contribuição para o campo da música electroacústica.*

Liduino Pitombeira *(Brasil, 1962) recebeu seu PhD em Composição pela Louisiana State University (E.U.A.), onde estudou com Dinos Constantinides. A sua música foi executada pelos seguintes grupos instrumentais: Quinteto de Sopros da Filarmônica de Berlim, Louisiana Sinfonietta, Red Stick Saxophone Quartet, New York University New Music Trio, Orquestra Sinfônica do Recife (Brasil), Syntagma, Orquestra Filarmônica de Poznan (Polonia) e Orquestra Sinfônica do Estado de São Paulo (Brasil).*

Pedro Patrício *é Compositor de Música Electrônica Contemporânea Erudita e Performer. É Doutorando em Ciência e Tecnologia das Artes: especialização em Informática Musical na Escola das Artes da Universidade Católica Portuguesa.*

Table of contents

Technical Papers

Music Papers

Posters

12°

Simpósio Brasileiro de Computação Musical

12th Brazilian Symposium on Computer Music

TECHNICAL PAPERS

Multidimensional microtiming in Samba music

Luiz Naveda[1], Fabien Gouyon[2], Carlos Guedes[3], Marc Leman[1]

[1]Institute for Psychoacoustics and Electronic Music - Ghent University
Blandijnberg 2, Ghent, Belgium

[2]Instituto de Engenharia de Sistemas e Computadores do Porto
INESC Porto

[3]Escola Superior de Música e das Artes do Espectáculo
Instituto Politécnico do Porto

luiznaveda@gmail.com, fgouyon@inescporto.pt, carlosguedes@mac.com, marc.leman@ugent.be

Abstract. *The connection of "groove" with low-level features in the audio signal has been mostly associated with temporal characteristics of fast metrical structures. However, the production and perception of rhythm in Afro-Brazilian contexts is often described as a result of multiple experience flows, which expands the description of rhythmical events to multiple features such as loudness, spectrum regions, metrical layers, movement and others. In this study, we analyzed how the microtiming of samba music interacts with an expanded set of musical descriptors. More specifically, we analyzed the interaction between fast timing structures with meter, intensity and spectral distribution within the auditory domain. The methodology for feature detection was supported by a psychoacoustically based auditory model, which provided the low-level descriptors for a database of 106 samba music excerpts. A cluster analysis technique was used to provide an overview of emergent microtiming models present in the features. The results confirm findings of previous studies in the field but introduce new systematic devices that may characterize microtiming in samba music. Systematic models of interactions between microtiming, amplitude, metrical structure and spectral distribution seem to be available in the structure of low-level auditory descriptors used in the methodology.*

1. Introduction

The connection of "groove" with low-level features of the audio signal has always been associated with the detection of rhythmical events and more specifically with the temporal characteristics of fast rhythmical structures. It has been suggested that the sensation of groove may be induced by small idiomatic variations of these rhythms, defined as a series of event shifts at a constant tempo (Bilmes 1993; Desain and Honing 1993; Gouyon 2007), or simply *microtiming*. In this study, we concentrate on the microtiming aspect of samba music, and how timing interacts with meter, intensity and spectral distribution.

Although the word "groove" may be closely related with music styles originating from the African-American diaspora, the induction of the groove feeling is also a common element in other musical contexts. Hennessy (2009) studied the groove in Cape Breton fiddle music (Canada) from the perspective of rhythmical formulas. Johansson (2005) studied microtiming and interactions with melodic patterns in Norwegian traditional fiddle music. Friberg and Sundstrom (2002) verified that eight-notes patterns are systematically performed in long-short patterns in jazz performances. The notion of swing in jazz and its correlations with pitch and phrasing was also studied in detail by Benadon (2003; 2006; 2009). Other studies tried to understand the notion of groove in different styles. McGuiness (2006) analyzed microtiming in different styles of music. Madison (2006) studied the consistence of the subjective grooving experience among subjects using music styles such as jazz, samba, Indian, Greek and Western African Music.

Recent studies have been concentrating on the characteristics of microtiming in Afro-Brazilian musical contexts. Gouyon (2007) analyzed the patterns of deviations of 16th-notes in samba-de-roda using a computational approach and a dataset of commercial recordings. Lindsay and Nordquist (2007) measured the microtiming of recordings of samba instruments using standard spectrograms. Part of the extensive study of Lucas (2002) about the *Congado Mineiro* was dedicated to the analysis of microtiming, based on field recordings in Minas Gerais. Gerischer (2006) connected several descriptions of the context of samba in Bahia with a systematic analysis of microtiming based on field recordings. Curiously, all of these studies describe systematic anticipations of the 3rd an 4th 16th-notes, which may configure a strong aspect of the Afro-Brazilian music styles.

1.1. Multidimensionality in Afro-Brazilian contexts

Gerischer claims that the rhythmical experience in samba should be understood as a multidimensional process based on oral traditions (Gerischer 2006, p.115) and aligned with the characteristics of African and Afro-Brazilian roots. Indeed, the culture of samba shares and incorporates the "*coordination of multiple experience flows*" (Stone 1985) of Afro-Brazilian rituals, which are claimed to be at the root of Afro-Brazilian music (Sodré 1979; Carvalho 2000; Fryer 2000; Sandroni 2001). A typical description of *Candomblé* ceremony demonstrates how these dimensionalities interact with each other in a certain context, and how the context is influenced by the musical experience:

> *"The dancers dance with great violence, energy, and concentration. Getting really involved in the rhythm and movement…The drummers… can play certain signals in the rhythmic pattern to cause the dancing to take a violent turn … One method is for one drum to syncopate the rhythm slightly (another one maintaining it) such that a strong beat falls just before the main beat…. This gives a impression of increased speed when this is not really the case, and creates tension and feeling of imbalance in the listener or dancer"* (Walker 1973; quoted by Fryer 2000)

In this ritualistic interaction between dance and music, music seems to be induced or induces a connection between timing and accents, a system of metrical levels, polymetric lines, instrumental textures, and a systematic mechanism of tension that provokes movement and cohesion. The musical elements in the samba culture seem to have inherited this structure of rituals and same multidimensionality. This aspect may be essential to describe rhythmic experience in Afro-Brazilian contexts or other cultural contexts influenced by African diaspora.

In this paper, we tried to investigate part of this multidimensional description of rhythmic experience using a systematic method based on computational approaches. We developed a methodology that describes interactions between timing, metrical levels, intensity and spectral distributions from musical audio. The methods include a psychoacoustically inspired feature detection (section 2.2.1) and a heuristic for microtiming detection (Section 2.2.2). The set of multidimensional descriptors of microtiming are extracted from a database of 106 music excerpts, and are then clustered using machine learning methods (section 2.2.3). By using these procedures, we aim at providing an overview of multidimensional interaction between microtiming and other mentioned features, which may help to uncover the elements of groove induction and thus improve the study of music forms within the Afro-Brazilian context.

2. Methodology

2.1. Data set

The dataset analyzed in this study consists of 106 excerpts of music collected from commercial CD's (median of durations = 33 seconds). The range of genres covered by this sample includes music styles influenced by Rio de Janeiro's samba, such as *samba carioca, samba-enredo, partido-alto and samba-de-roda* (Bahia). The excerpts were stored in mono audio files with a sample rate of 44.100 Hz / 16 bits and normalized by amplitude. Beat markers and the metrical positions of the first annotated beat (1^{st} or 2^{nd} beat, 2/4) were manually annotated by 3 specialists using the software Sonic Visualizer (see Cannam, Landone et al. 2006).

2.2. Analysis

Our analysis was developed in 3 stages: (2.2.1) definition of low level features and spectral regions, (2.2.2) segmentation of metrical structures and extraction of event features, and (2.2.3) clustering of multidimensional information.

2.2.1 Definition of low level features and spectral regions

2.2.1.1. Auditory model

In order to provide an robust low-level feature for the representation of musical tessitura we used an implementation of the auditory model described in Van Immerseel and Martens (1992), implemented as a .dlib library for Mac OSX. This auditory model simulates the outer and middle ear filtering and the auditory decomposition in the periphery of the auditory system. This results in loudness patterns distributed over the audible spectrum (for more details see Van Immerseel and Martens 1992, p. 3514). The configuration used in this study provides 44 channels of loudness curves with sample frequency at 200 Hz, distributed over 22 critical bands (center frequencies from 70 Hz to 10.843 Hz). Figure 1 displays an auditory image (or loudness curves) generated from the auditory model of the excerpt 22.

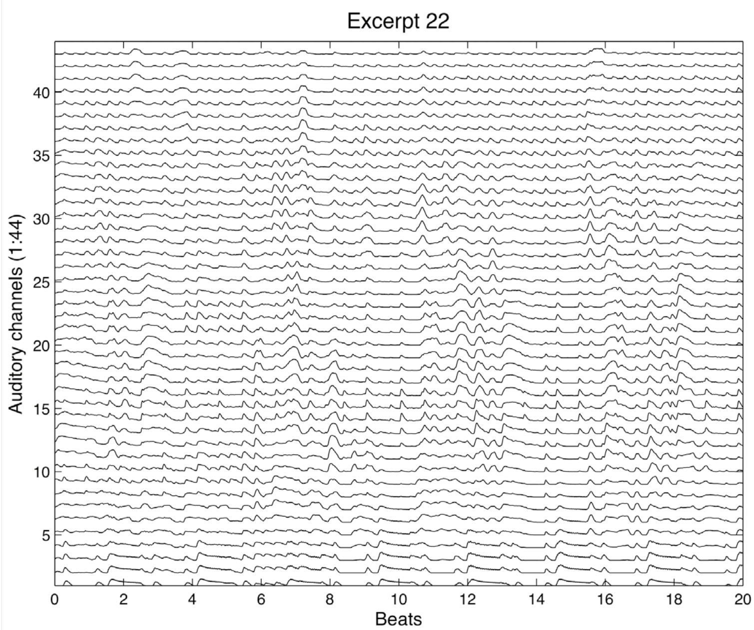


Figure 1. Loudness curves generated from the auditory model. The 44 envelope curves (0:1 x 44 channels) represent a simulation of loudness in each auditory channel.

2.2.1.2. Segmentation of spectral distributions

The current knowledge about samba forms accounts for relatively stable configurations of musical instruments (and their musical functions) across the musical tessitura and the musical function of each instrument is often related with its timbre. Timbre can be roughly represented by low-level descriptors in the frequency domain or, in our case, by loudness amplitudes in time distributed in auditory channels. The spectrum of the low bass samba percussion, *Surdo,* is mostly concentrated in the lower part of the audible spectrum. *Tamborims*, *repiniques*, vocal parts and other instruments occupy the mid frequency region of the auditory spectrum. *Ganzás* and different kinds of shakers will tend to occupy the higher spectrum regions. Although the frequency components of these instruments will overlap each other in the time and frequency domain (particularly during transients at attacks points), the spectrum signature of each timbre is relatively discriminated from each other. Figure 2 displays the analysis of the spectrum of four kinds of samba instruments and the distribution of central frequencies of the auditory model within the audible spectrum.

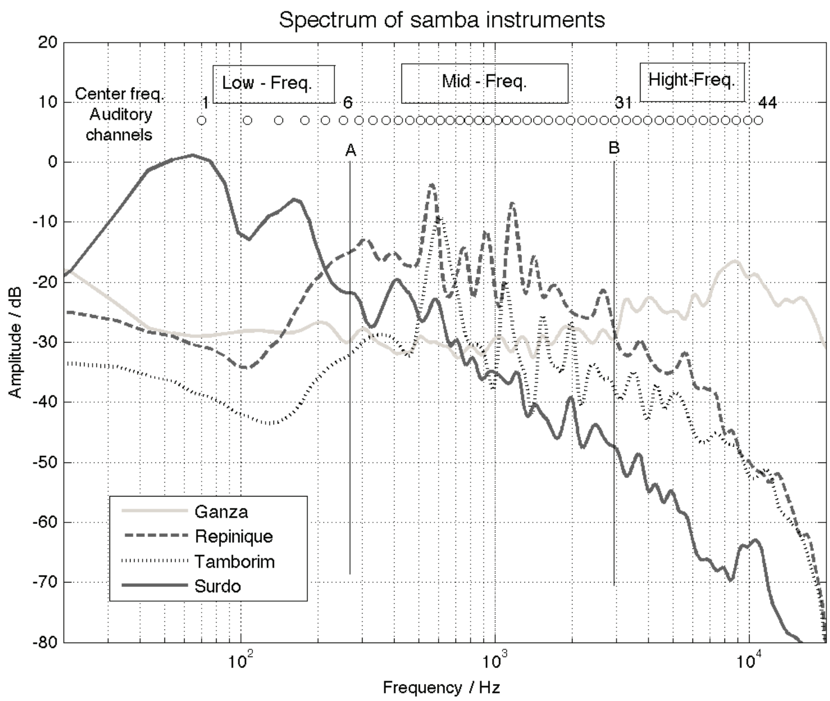


Figure 2: Spectrum analysis of a typical samba percussion set ensemble (hamming window, 4096 points, 1/5 octave smoothing). The distribution of the central frequencies of the auditory channels across the frequency domain is indicated by the [o] marks (arbitrary amplitude). The divisions A and B indicate the segmentation of the auditory channels in 3 groups (low, mid and high).

Each sound excerpt was processed by the auditory model, which resulted in 44 loudness curves (44 auditory channels). The loudness curves were averaged in 3 loudness curves that reflect estimated distributions of tessitura: low-frequency region – channels 1:6, mid-frequency region – channels 7:30 and high-frequency region - channels 31:44 (for a similar procedure see Lindsay and Nordquist 2007).

2.2.2. Segmentation of metrical structures and extraction of features

The interactions between features may differ if different metrical layers are taken into consideration, which means that variations of timing, amplitude and spectrum may change if observed in relation with musical beats or bars. We define metrical levels as a set of hierarchical levels that can be operationally represented by multiples and divisions of the beat positions (e.g.: 1-beat, 2 beats, 1/2 beat). Current knowledge about the samba forms indicates that samba music has a well-defined beat level, consisting of a binary bar structure (2 beats) and a fast metrical onset structure at a mathematical ratio of ¼ of the beat (known as *tatum* layer). Each metrical element of the microtiming level will be referred to as 16^{th} notes (mathematical subdivision of 1/4 of the beat). The shifts of the mathematical position of the 16^{th}-notes will be described in relation with 2 macro levels, namely the beat (1-beat) and bar (2-beat) levels.

The annotation of beat and bar positions is essential for the segmentation of metrical structures. Manual beat annotation provides a proper human evaluation of the beat points but lacks precision at a microtiming level. Automatic beat annotation is precise at microtiming level and relies on the use of a systematic rule in order to find the beat positions. However, the analysis of samba music with software such as Beatroot (Dixon

2007) and Sonic Visualizer (Cannam, Landone et al. 2006) resulted in poor beat tracking results, probably influenced by the characteristic rhythmic complexity of the samba music. Therefore, we opted to combine manual and automatic approaches in a heuristics that looks for relevant peak events in the proximity of manual annotation. The applied method is described below and in the Figure 3.

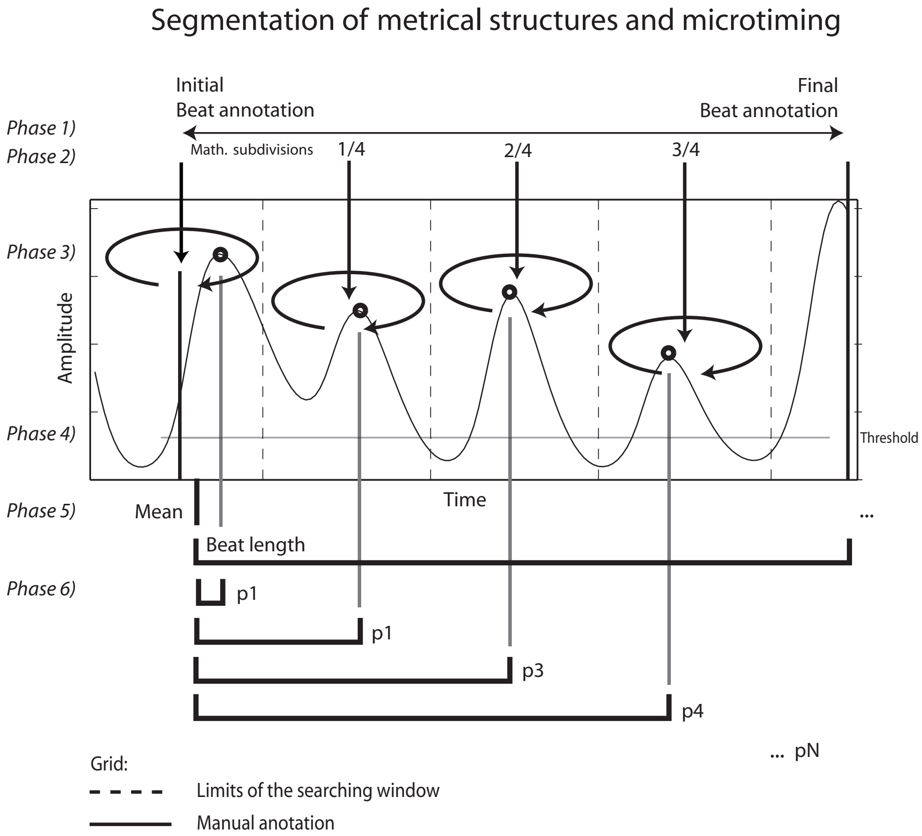


Figure 3: Description of the heuristic used for segmentation. See text below for the explanation of each step.

Description of the algorithm:

For each excerpt, for each spectral region, for each metrical level,

Phase 1. Retrieve beat points and time interval of the actual metrical segment from the manually annotated beats (e.g.: inter beat interval, inter-bar interval).

Phase 2. Project the mathematical divisions of the microtiming points, here defined as ¼ of the beat length (e.g. inter-beat interval/4)

Phase 3. Look for the peaks in the proximity of/ within the range of manual annotation (length of the window = microtiming period)

Phase 4. Select a higher peak situated above a determined threshold (if there are no peaks above threshold, retrieve NaN).

Phase 5. Extract the mean peak position of the first peaks of the 3 spectral regions. Therefore, all positions (including positions in different spectral regions) have the same beat reference.

Phase 6. Retrieve position and amplitude of the highest peak in close proximity of the mathematical subdivisions.

Phase 7. Retrieve features: (A) the normalized length in relation with the length of metrical layer, (B) peak amplitudes, metrical levels (C) and spectral regions (D)

The result of this process is a multidimensional feature description of microtiming, represented by four kinds of indicators: (A) the position of the peak in relation with the metrical layer, (B) the intensity of the peaks, (C) the region in the spectra and (D) the metrical level.

The definition of the length of the metrical level is crucial for the projection of microtiming ratios. However, the definition of precise beat length information is not trivial. Apart from the problems with automatic and manual beat annotation, first peak positions are often different for each region of the spectra. In order to provide a referential point, the beat position is considered as the mean between the first peak positions of the three spectral regions (Phase 5).

2.2.3 Clustering of multidimensional information

The information resulting from the feature detection is composed of (1) the ratios between all peak positions and the metrical level length, (2) amplitude of the peak (loudness curves), (3) metrical level and (4) spectral region (low, mid, high). In order to find trends and interactions between these feature components, we carried out a *k-means* clustering based on an improved extension of the basic *k-means algorithm* (Pelleg and Moore 2000). We configured the algorithm to retrieve a minimum of 3 and a maximum of 5 clusters and it was implemented in Weka platform (Witten and Frank).

3. Results

The results are displayed for metrical levels 1-beat and 2-beat. In this study, peak positions are indicated as 16th-note positions. These positions represent the subdivision of ¼ of the beat.

The cluster representations provide visual information about mathematical division of the metrical levels (grids) and cluster affiliation. Different stem markers represent different clusters. Ticks distributed along the horizontal axis have a resolution of 0.05 beats.

3.1. Metrical layer: 1-beat

The analysis of the 1-beat level resulted in three clusters displayed in Figure 4. The representation of the cluster centroids shows a systematic anticipation of 3rd and 4th peaks in all three frequency regions and in all three clusters. The first 16th-note of the low-region is slightly delayed, especially in the cluster that shows higher energy c1-[o]. The second 16th-note seems to be accentuated in the mid- and high-region of the clusters c1-[o] and c2-[x]. The high portion of the signal shows flat amplitudes in second half of the beat in all clusters.

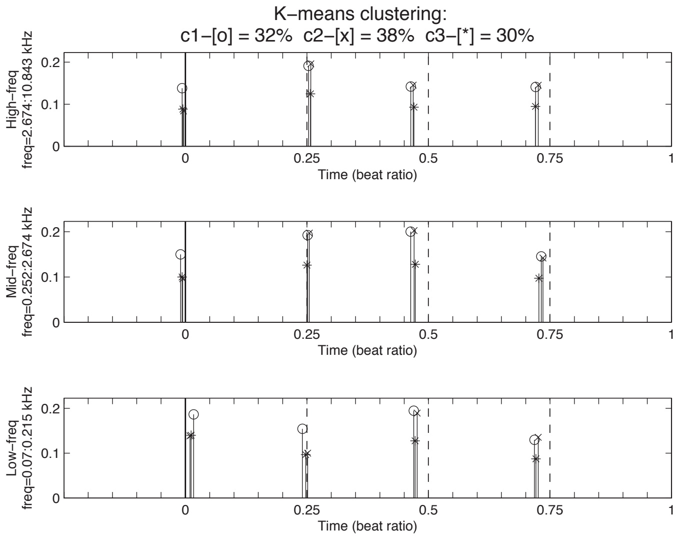


Figure 4: Representation of centroids for 3 clusters and 5064 instances (beats). Ranges of tick periods (0.05 beats) for each cluster: c1= 22:45 ms, c2= 20:52 ms, c3= 22:52 ms.

The database comprises a wide range of tempi, which implies that temporal information represented in beat ratios denotes different temporal ranges in seconds. As for the 3rd and 4th peaks, variations in Figure 4 seem to be greater than 0.025 beats but smaller than 0.05 beats, which indicates a range of anticipations between 10 and 52 ms. In the case of the 1st 16th-note (low-region), the delay seem to correspond to a period smaller than 0.025 beats, or 21 ms (at the slower tempi).

3.2. Metrical level: 2 beats

The clusters of the metrical level 2-beat offer a broad overview of microtiming relations at the 2-beat (bar) level. The clustering process resulted in five clusters displayed in Figure 5 and Figure 6.

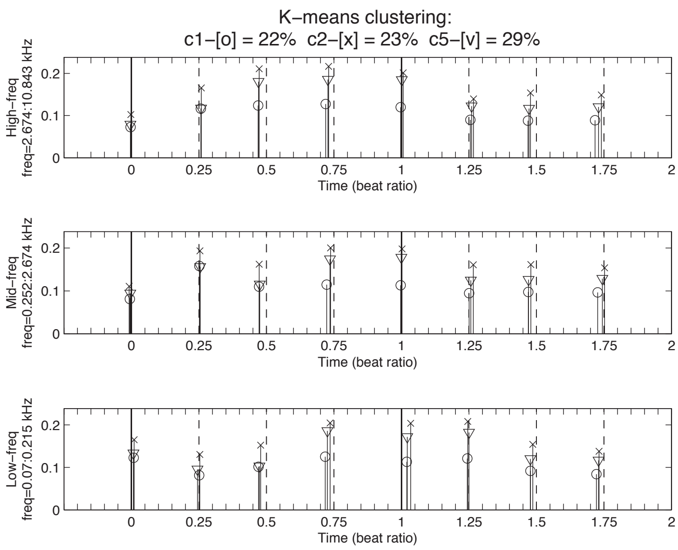


Figure 5: Representation of cluster centroids for 3 clusters and 1859 instances (clusters 1,2 and 5, 2-beats). Ranges of tick periods (0.05 beats) for each cluster: c1= 22:52 ms, c2= 20:45 ms, c5= 22:45 ms.

Figure 5 shows clusters 1, 2 and 5. The results show the same systematic anticipations of 3rd and 4th 16th-notes and a delay of the 1st 16th-note in the low-frequency region. These microtiming deviations seem to affect the two beats at the bar level and show the same temporal range at the metrical level 1-beat. In addition, the delay of the first 16th-note (low-frequency) seems to be more significant in the second beat. However, this delay has a broader range, situated between 11 ms (for the fastest tempi in c1) and 45 ms (for the slower tempi of c2).

Peak amplitudes reveal more variability at this metrical level. While the peak of second 16th-note seems to be accentuated only in the mid-frequency region (1st beat), the fourth 16th-note is accentuated in the clusters 2–[x] and 5-[v]. However, in the 2nd beat, peak amplitudes of the 2nd to the 4th 16th-notes are flattened.

Figure 6 shows the results of the clusters 3 and 4. These results differ from the clusters displayed before because they show increasing deviations accumulated along peak positions.

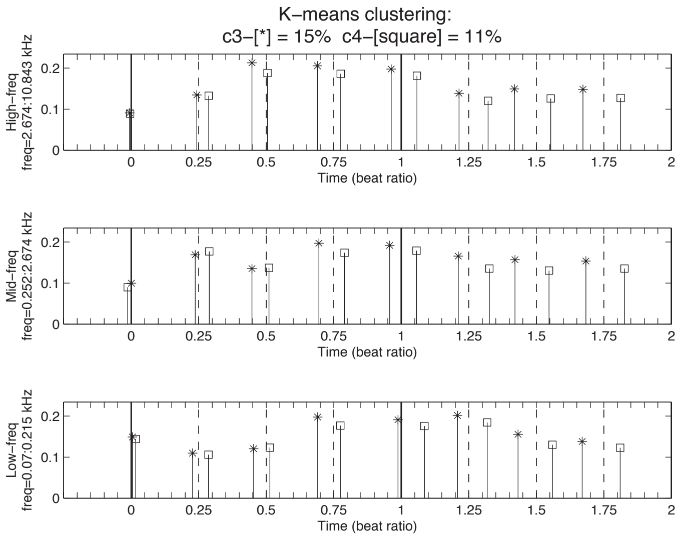


Figure 6. Cluster centroids for 2 clusters and 659 instances (clusters 3 and 4, metric level 2-beats). Ranges of tick periods (0.05 beats) for each cluster: c3= 20:45 ms, c4= 20:52 ms.

Cluster c3-[*] shows an increasing anticipation in all regions and peaks. The anticipation increases until the last 16th-note of the 2nd beat, which shows an anticipation of almost 0.1 beat (from the mathematical rule At 1.75 beats). Cluster c4-[square], shows the opposite behavior, displaying a crescent delay, from the first to the last 16th-note. A clear delay of the 1st 16th-note in the low-frequency region can also be observed. The amplitude patterns seem to be similar to the observed amplitudes in clusters c1, c2 and c5.

4. Discussion

The systematic recurrence of anticipation in the 3rd and 4th 16th-notes in all metrical levels and spectral regions seem to confirm the existence of a systematic artifact described in previous studies about microtiming in samba music (Gerischer 2006; Gouyon 2007; Lindsay and Nordquist 2007) and other Afro-Brazilian traditions (Lucas 2002). Variations of these peak positions seem to be greater than 0.025 beats but smaller than 0.05 beats. All this is situated within a range of anticipations between 10 and 52 ms of the mathematical division of the beat (0.5 and 0.75 beats).

The systematic delay of 1st 16th-note positions in the specific low frequency region of the spectrum for all metrical layers shows an observation not mentioned in previous studies. It is well know that low-frequency spectrum is often dominated by commetric beat patterns, performed by percussion instruments such as *surdo* or *tantan*, and that these bass lines are often accentuated in the 2nd beat (Chasteen 1996; Sandroni 2001; Moura 2004), which also seem to be reflected in our results. However, we were unable to find references to any systematic delay of bass percussion instruments.

The delay of 1st 16th-note positions must be interpreted attentively. The temporal range of delays in the low frequency region is very close to the sample period of the auditory model (5 ms), which means that minimum significant delays found in the Figure 4, for example, account for only 2-samples (10 ms) between the mathematical rule and peak position. More research is needed to support this observation.

The occurrence of linear and crescent deviations, demonstrated in Figure 5, must be also interpreted with care. The computation of clusters may have merged two recurrent tendencies of outliers in the data set. However, the magnitude of instances represented by these clusters (c3-15% and c4-11%) and similar cluster structures found in other metrical levels above 2-beats (4-beats level, not shown in this study), indicate that they reflect real microtiming structures represented in our data-set. If this hypothesis is confirmed, the presence of these clusters may be attributed to rhythmic devices similar to *accelerando* and *ritardando* forms. Although these rhythmical artifacts are widely used to delimit phrases, endings and formal articulations in classical music, it is surprising that such devices appear in our dataset. The range of these deviations indicate that they are less clearly defined than the ones used in classical music, which may configure a new microtiming device.

The variation of amplitudes demonstrate that microtiming in samba is subjected to interactions with accents and meter. The flatness of 16th-note amplitudes observed in clusters in all metrical levels, especially the 2-beat level, indicate the existence of metrical cues encoded in the amplitude of microtiming structures. While the first beat starts with a low-energy 16th-note in the low-frequency region and accents in the 2nd (Figure 4) and 4th peaks (Figure 5), the 2nd beat starts with a characteristic strong bass accent, followed by flat and low intensity 16-th notes. This oscillation of multidimensional characteristics between beat positions may play an important role in the induction of grooving and reinforce metrical properties.

5. Conclusion

In this study we analyzed the interaction between microtiming, meter, intensity and spectrum. The results strongly confirm the systematic tendency of anticipations of the 3rd and 4th 16th-notes at the metrical level of 1 beat. It also shows the presence of two new rhythmic devices that may characterize samba forms: (1) a small delay of the bass lines and (2) systematic forms of *acelerando* and *ritardando* at a microtiming level. Peak amplitudes seem to work according to two functions: (1) the induction of systematic accents in the 3rd and 4th 16th-notes of the first beat (metrical level 2-beats) and (2) an artful mechanism that interacts with energy between metrical structures and spectral regions. The use of a psychoacoustically based feature as a low-level descriptor suggests that these observations are available as proximal cues in the periphery of the auditory system. Moreover, the results show that microtiming can be understood as a multidimensional device of musical engagement.

The present study does not intend to show an exhaustive overview of multidimensionality of microtiming structures in Afro-Brazilian music. Other important interactions inside and outside the auditory domain may influence the process. In addition, more work is needed to elucidate the role and the magnitude of these findings within the perception of groove induction.

5. Acknowledgements

This study was supported by a Short Term Scientific Mission, COST – STSM IC0601-3753, by a grant from Ghent University (Belgium) and partly CAPES (Brazil). The authors wish to thank Inaê Benchaya Duarte for the annotation files.

References

Benadon, F (2003). The expressive role of beat subdivision in jazz. Conference Proceedings of the Society of Music Perception and Cognition, Las Vegas.

Benadon, F (2009). "Time Warps in Early Jazz." Music Theory Spectrum **31**(1): 1-25.

Benadon, F. (2006). "Slicing the beat: Jazz eighth-notes as expressive microrhythm." Ethnomusicology **50**(1): 73-98.

Bilmes, Ja (1993). Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm, Massachusetts Institute of Technology.

Cannam, C, C Landone, M Sandler and Jp Bello (2006). The sonic visualiser: A visualisation platform for semantic descriptors from musical signals.

Carvalho, J. J. De (2000). Afro-Brazilian Music and Ritual [s], Duke-University of North Carolina Program in Latin American Studies.

Chasteen, Johncharles (1996). "The prehistory of Samba: Carnival Dancing in Rio de Janeiro, 1840-1917." Journal of Latin American Studies **28**(1): 29-47.

Desain, P and H Honing (1993). "Tempo curves considered harmful." Contemporary Music Review **7**(2): 123-138.

Dixon, S. (2007). BeatRoot 0.5.6.

Friberg, A and A Sundstrom (2002). "Swing ratios and ensemble timing in jazz performance: Evidence for a common rhythmic pattern." Music Perception **19**(3): 333-349.

Fryer, P. (2000). Rhythms of Resistance: African Musical Heritage in Brazil, Pluto.

Gerischer, C. (2006). "O suingue baiano: Rhythmic feeling and microrhythmic phenomena in Brazilian percussion." Ethnomusicology **50**(1): 99-119.

Gouyon, F. (2007). Microtiming in "Samba de Roda"—Preliminary experiments with polyphonic audio. XII Simpósio da Sociedade Brasileira de Computação Musical. São Paulo.

Hennessy, J (2009). Fiddle Grooves: Identity, Representation, and the Sound of Cape Breton Fiddle Music in Popular Culture. Department of Music. Toronto, University of Toronto. **Doctor of Philosophy**.

Johansson, M (2005). Interpreting micro-rhythmic structures in Norwegian traditional fiddle music. Rhythm and Micro-rhythm: Investigating musical and cultural aspects of groove-oriented music. Oslo.

Lindsay, K. and P. Nordquist (2007). "Pulse and swing: Quantitative analysis of hierarchical structure in swing rhythm." The Journal of the Acoustical Society of America **122**: 2945.

Lucas, G. (2002). "Os sons do Rosário: o congado mineiro dos Arturos e Jatobá." Belo Horizonte: Editora UFMG.

Madison, Guy (2006). "Experiencing Groove Induced by Music: Consistency and Phenomenology." Music Perception **24**(2): 201-208.

Mcguiness, A (2006). Microtiming deviations in groove. Centre for New Media Arts. Canberra, Australian National University.

Moura, R. M. (2004). No princípio, era a roda: um estudo sobre samba, partido-alto e outros pagodes, Rocco.

Pelleg, D and Aw Moore (2000). X-means: Extending K-means with efficient estimation of the number of clusters, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Sandroni, C. (2001). Feitiço decente: transformações do samba no Rio de Janeiro, 1917-1933, Jorge Zahar Editor: Editora UFRJ.

Sodré, Muniz (1979). "Samba, O Dono do Corpo." Rio de Janeiro: Codecri.

Stone, Rm (1985). "In Search of Time in African Music." Music Theory Spectrum: 139-148.

Van Immerseel, Lm and Jp Martens (1992). "Pitch and voiced/unvoiced determination with an auditory model." The Journal of the Acoustical Society of America **91**: 3511.

Walker, Ss (1973). Ceremonial spirit possession in Africa and Afro-America: Forms, meanings, and functional significance for individuals and social groups, Brill Archive.

Witten, Ih and E Frank "WEKA Software, v3. 5.2." University of Waikato.

Interacting with the 3D Reactive Widgets for Musical Performance

Florent Berthaut[1] **, Myriam Desainte-Catherine**[1] **, Martin Hachet**[2]

[1]Université de Bordeaux
Laboratoire Bordelais de Recherche en Informatique
351 cours de la libération – 33400 Talence, France

[2]INRIA
Laboratoire Bordelais de Recherche en Informatique
351 cours de la libération – 33400 Talence, France

berthaut@labri.fr, myriam@labri.fr, hachet@labri.fr

Abstract. *While virtual reality and 3D interaction provide new possibilities for musical applications, the existing immersive virtual instruments are limited to single process instruments or musical navigation tools. In this paper we present the 3D reactive widgets. These graphical elements enable simultaneous control and visualization of musical processes in 3D immersive environments. They also rely on the live-looping technique, allowing to build complex musical sequences. We describe the interaction techniques that we have designed and implemented to manipulate these widgets, including a virtual ray and tunnels. After having expressed the lack of expressivity and efficiency of the existing input devices for sound production gestures, we finally set the requirements for an appropriate device for musical interaction in 3D immersive environments.*

1. Introduction

Graphical musical interfaces have many advantages over hardware controllers. They can provide easy and direct access to a high number of parameters of an unlimited number of sound processes. They can also be used to display many useful informations about these processes. Nevertheless, they require efficient interaction devices and techniques.

The purpose of our work is to explore the possibilities provided by 3D interaction and virtual environnements. Navigation in these environments can be a good metaphor for navigation in musical pieces or exploration of musical structures. New interaction techniques developped in these fields of research open possibilities for musical interaction and can also be combined with traditional techniques. Immersion also provides new sensations to musicians and to the audience. However, these possibilities must be adapted to the specific needs of musical interaction, such as expressivity, efficiency, and minimum latency.

In this paper, we present the principle of the 3D reactive widgets. Then we describe the specific interaction techniques that we have developped. We finally set the requirements for an input device for 3D musical interaction.

2. Related Work

Relatively few research have been done in the field of 3D interaction and virtual reality for music. Some of them focus on navigation in musical environments, like the virtual groove in the Phase project [Rodet et al., 2005], or the audiovisual grains in Plumage [Jacquemin et al., 2007]. The applications developped by Mike Wozniewski et al. [Wozniewski et al., 2006] also rely on users movements to either control the spatialization of pre-recorded sound sources, or apply effects on the sound of an accoustic instrument. Some immersive instruments are single processes instruments, i.e. instruments that allow to interact with only one synthesis process, such as the Virtual Xylophone, the Virtual Membrane, or the Virtual Air Guitar developped by Mäki-Patola et al. [Mäki-Patola et al., 2005] and the sculpting instruments developped by Mulder [Mulder, 1998]. Finally, among the existing multi-processes 3D instruments, part of them, like the WAVE software from Valbom et al. [Valbom and Marcos, 2005] or the application developped by Martin Naef et al. [Naef and Collicot, 2006], have limited visual feedback and interaction possibilities since they tend to emulate hardware controllers. The other part of these instruments rely on gaming software or devices, like the 3D instruments Fijuu [Olive and Pickles, a] and Ergates [McCormick,], or the collaborative musical First Person Shooters q3apd [Olive and Pickles, b] and q3osc [Hamilton, 2008]. They offer new interaction techniques and interesting visualizations, but they do not take advantage of the possibilities of immersive environments. None of these applications combine immersion, simultaneous control of multiple processes, expressive interaction techniques and complex visual feedback.

3. The 3D Reactive Widgets

3.1. Principle

Our research focus on using 3D immersive environments for musical interaction. These environments indeed add possibilities in terms of temporal or hierarchical navigation by means of 3D movements. They also enable the design of new interaction techniques and paradigms, for example using the additional dimensions for manipulation and visualization. Furthermore, immersion can improve the experience of the musicians who will better perceive and thus manipulate the 3D interface with stereoscopic display and head-tracking. But it will also improve the experience of the audience, if they are equipped with stereoscopic glasses, both for the spectacular aspect and for the understanding of the musicians playing.

As said in the previous section, most virtual reality instruments are single process instruments, i.e instruments that allow the control of only one synthesis or effect process, or musical navigation tools. However, we believe that the main advantage of graphical musical interfaces is to give the possibility to handle multi-processes instruments with control on and visual feedback from the selected sound processes.

This is why we chose to rely on the concept of reactive widgets described by Golan Levin[Levin, 2000] and used for example by Sergi Jorda[Jordà, 2005] in FMOL. A reactive widget is a graphical component which allows both manipulation and visualization of a musical process. Its graphical parameters are connected to the parameters of the associated musical process. These connections are bidirectionnal, so that graphical changes are reflected in the sound process and that musical events are displayed in return by the widget. The efficiency of this concept lies in the shortening of the "indirection degree" described by Michel Beaudoin-Lafon[Beaudoin-Lafon, 1999] because there is direct manipulation of the "objects of interest", in our case the visualized sound processes.

These observations led us to adapt the concept of the reactive widgets to 3D immersive environments, as it can be seen in figure 1.

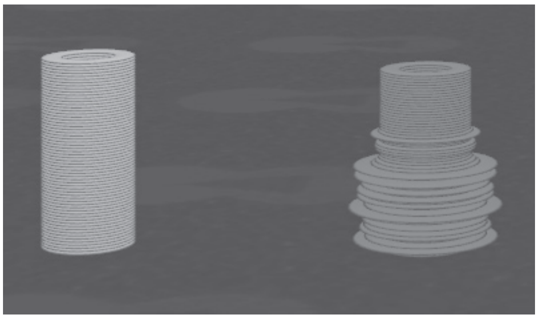


Figure 1: Two 3D Reactive Widgets: the one on the right shows the spectrum of its associated sound process.

3.2. Audiovisual Mappings

Several questions emerge from the concept of the 3D reactive widgets. First of all, one must choose which sound and musical parameters should be controlled and visualized. Symetrically, one must choose which graphical parameters should be used to manipulate and display the sound processes. Finally, one must choose the mappings between these audio and graphical parameters. These questions have been explored in several user studies, such as [Giannakis, 2006], but always in terms of users preferences and with a music composition perspective. This is why we are currently running a specific user study aimed at interaction, which will focus on mappings efficiency by measuring subjects' performances. Meanwhile, the following parameters and mappings were chosen for our first implementation: spectrum with shape, color lightness with pitch, size with loudness, shape distortion with brightness, and transparency with noisiness.

3.3. Sound processes

The sound process attached to each 3D reactive widget is composed of a sample player, an audio effects rack and an audio analysis rack. The widget, when activated by a specific manipulation, triggers the sample. The sound goes through the effects rack, whose parameters are linked to the graphical parameters of the widget according to the mappings described in the previous subsection. The result is then sent to the soundcard output, and at the same time analyzed to set the widget's shape. A very important feature of our application is that the sound triggers and the manipulations of the graphical parameters can be recorded and looped with the live-looping technique. This technique is used by a growing number of musicians in every musical genre and relies on music software or hardware devices. In our case, when the musician grabs a widget, he or she can press a button to start recording. All the following parameters variations are recorded, without temporal quantization, and looped when the button is pressed again. All the loops of a widget are synchronized, and can easily be deleted. This allows to build, stack and manipulate complex sound processes using the 3D reactive widgets.

4. Implementation

In the current implementation, as it can be seen in figure 3, the musician is equipped with head-tracked stereoscopic glasses and uses tracked Wii Remotes. The display is a large screen combined with Infitec stereo projectors and the tracking is done with the A.R. Tracking 6DOF DTrack system[1]. A cheaper system could be set up with simple anaglyph stereoscopy combined with tracking using a wiimote IR sensor. Tracking data is transmitted using VRPN[2].

[1]http://www.ar-tracking.de/
[2]http://www.cs.unc.edu/Research/vrpn/

The 3D environment is rendered by an application which we have developped, called Poulpe. Poulpe is based upon the OpenSG scenegraph library[3] and the oscPack library[4]. It can be seen as a bridge between virtual reality and music software because it associates musical control messages with each parameter of each element of a 3D scene. Thus the camera, the lights and the 3D objects can be used to control and visualize musical processes. The scene is described in a text file with an xml syntax.

In our application, Poulpe communicates with custom music software using the OpenSoundControl[5] protocol. The music application uses the Jack sound server[6]. Sound files are associated with each 3D reactive widget. Effects are applied using LV2 plugins[7]. Finally, audio analysis is done with Vamp plugins [8]. The use of plugins allows to easily add new effects and sound features. The resulting setup is shown in figure 2.

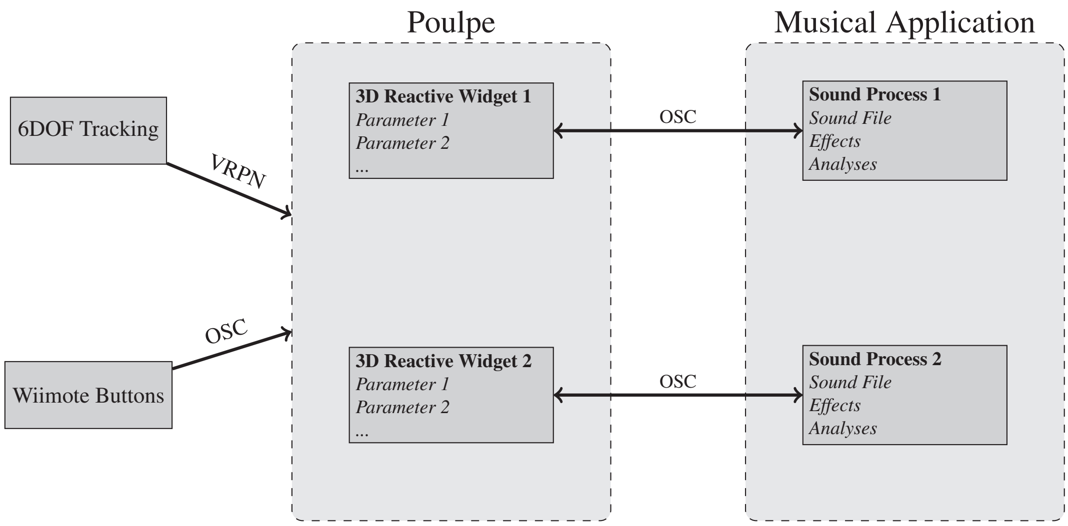


Figure 2: Block diagram of the current implementation.

5. Interacting with the 3D reactive widgets

In the next subsections, we describe the first interaction techniques chosen to manipulate the 3D reactive widgets. These techniques can be seen in figure 3 and in a video[9]. They can be categorized, by refering to Cadoz's [Cadoz, 1999] work on musical gestures, as selection gestures, modulation gestures and excitation gestures.

5.1. Virtual Ray

In order to select and to grab the reactive widgets, the musician manipulates a virtual ray, which is commonly used in virtual reality applications [Bowman et al., 1997]. It was evaluated as an efficient technique especially for near objects, by Poupyrev et al. [Poupyrev et al., 1998]. It gives a feeling of continuity from the real world and provides sufficiently accurate and fast pointing in the virtual environment. In our application, the movements of the ray are low-pass filtered when over a non-grabbed reactive widget, to

[3]http://opensg.vrsource.org/trac
[4]http://www.audiomulch.com/ rossb/code/oscpack/
[5]http://opensoundcontrol.org/
[6]http://jackaudio.org/
[7]http://lv2plug.in/
[8]http://www.vamp-plugins.org/
[9]http://rapidshare.com/files/249367217/interacting-3D-reactive-widgets-musical-performance.mpg.html

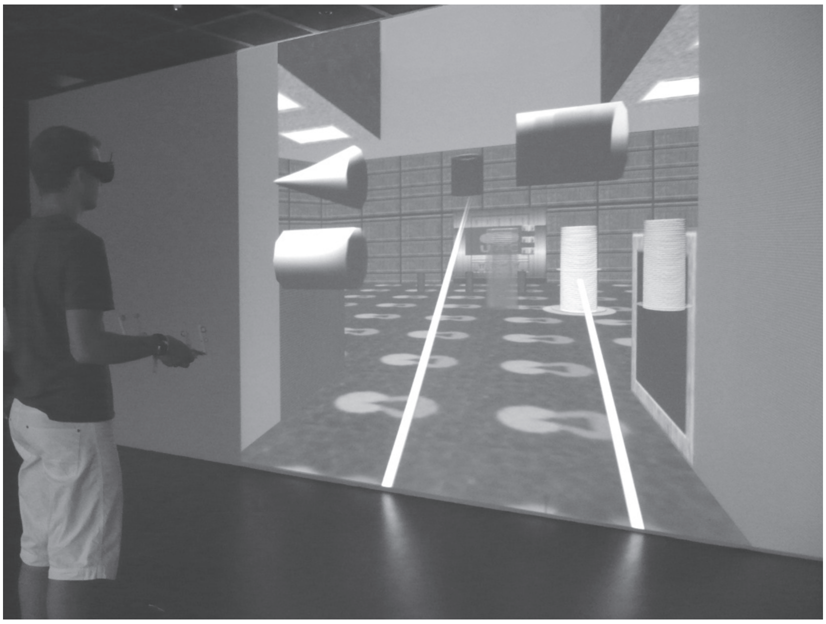


Figure 3: The current application with four 3D reactive widgets, three tunnels and two virtual rays.

avoid unwanted jumps while triggering the widget. We also chose to limit the translations applied to the widget to the X and Y axes to make the selection and manipulation gestures easier keeping the movements on the Z axis for navigation in the 3D environment. Two rays can finally be used at the same time, one in each hand. This technique corresponds to the selection gestures, by which musicians choose the instruments components, like the key on a piano.

5.2. Tunnels

In order to manipulate the widgets' graphical parameters, the musician moves them through what we call "tunnels". Each tunnel modifies one graphical parameter, and thus the corresponding sound parameter, following specific scales. Scales presets, such as non-linear or discretized scales, can be defined in the Poulpe configuration file and selected while playing. Our current application is thus composed of a Size Tunnel, a Distortion Tunnel, a Transparency Tunnel and a Color Tunnel. They can be seen in figure 4. As indicated in the previous section, these variations can be live-looped. This technique corresponds to the parametric modulation gestures, which modify the properties of an instrument, such as the note on a string of a violin.

5.3. Sounds Trigger

Finally, in order to trigger the sounds of the 3D reactive widgets, two techniques were experimented: the buttons of the wii remote and the collisions of the widgets (with vibro-tactile feedback). The buttons allow for simultaneous excitation and modulation gestures, but they lack expressivity and temporal accuracy. Triggering the sounds by hitting the widgets against other objects is more expressive, because the speed of the collision can be used. But, on the other hand, it increases the latency and requires more accurate active haptic feedback. These two techniques are thus not efficient enough to be used as the excitation gestures, i.e. gestures that directly and physically generate the sound. This was indeed confirmed during the demonstrations that we made for the VRST 2008 conference. In order to enable efficient musical interaction in immersive environments, a new device must be designed.

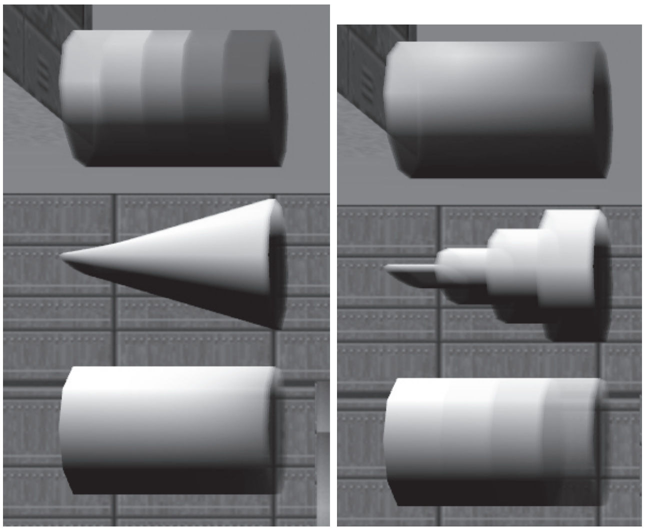


Figure 4: Three Tunnels: Color/Pitch, Size/Volume, Transparency/Noisiness. By sliding a 3D reactive widget through them, the musician modifies its graphical parameters, and thus the corresponding parameters of the associated sound process. The second figure shows a different preset for each tunnel.

6. Towards a new interaction device

6.1. Related devices and limitations

The interaction device currently used in our application is a wii remote equipped with 6DOF targets. This device, as other videogames controllers such as joysticks and gamepads, does not provide controls that are expressive and accurate enough. Moreover, the bluetooth protocol used for data transmission adds an average of 35ms of latency. Virtual reality devices, such as the flystick from A.R Tracking, focus on accurate orientation and position sensing and does not provide expressive controls. Advanced hardware instruments, for example the ones presented in the NIME conferences[10], usually focus on expressivity. However they are not usable for graphical interaction because either they are designed for single processes instruments, or they are not generic enough, or they are not handheld and limit the musicians movements. The Meta-Instrument[11] could be a good solution, since it provides a lot of expressive inputs and sensing of arms movements. However, it does not give absolute 3D position and orientation of the musician and the control of the virtual ray with the forearm would be far less accurate and fast than with the wrist. Advanced haptic controllers such as the one designed by the ACROE[12] restrict the movements in translation and rotation and thus are not desirable for interaction in immersive environments, especially for the control of a virtual ray. Finally, the few devices specifically designed for immersive musical interfaces like the device for Ashitaka [Moody, 2006] or the Sphere Spatializer [Wakefield et al., 2008] focus on navigation and do not enable expressive excitation gestures.

6.2. Requirements

Considering the limitations of existing devices, there is a need for an efficient device for musical interaction in immersive environments. We believe that it should meet the following requirements:

[10]http://www.nime.org/
[11]http://www.pucemuse.com
[12]http://acroe.imag.fr/ergos-technologies/index.php?idp=0

Figure 5: Using wii remotes with 6DOF tracking to control the virtual rays

- It should provide accurate and fast 6DOF tracking.
- It should allow simultaneous excitation and modulation gestures.
- The communication protocol used should minimize latency, so that percussive gestures can be correctly achieved.
- The number of controls available and their expressivity should be maximized.
- It should be easily handheld.
- It should be sufficiently generic to enable the design of new interaction techniques and should be usable for non-musical applications.
- Finally, it should provide haptic feedback (at least passively).

7. Conclusion

We described a new concept for graphical musical interaction: the 3D reactive widgets. Each of these widgets allow both the control and the visualization of a musical process. This principle is combined with the possibilities provided by 3D immersive environments in terms of interaction and immersion. It is also combined with the live-looping technique, allowing the creation of complex musical sequences. We introduced the first interaction techniques designed to manipulate these widgets, which rely on the virtual ray for the selection gestures, and on what we have called tunnels for modulation gestures. In the last section, we expressed the lack of an appropriate device for the excitation gestures, i.e. gestures that generate the sound. Further work will be to design this new interaction device and new techniques that it will enable. This device could also be used for other immersive applications, even non-musical ones that requires accuracy and expressivity. Finally, our in-progress user study on audiovisual mappings may help choosing the right mappings for various graphical musical interfaces, including our application.

References

Beaudoin-Lafon, M. (1999). *Interfaces homme-machine et création musicale*, chapter Moins d'interface pour plus d'interaction. Hermès Sciences.

Bowman, D., Koller, D., and Hodges, L. F. (1997). Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In *IEEE Proceedings of VRAIS97*.

Cadoz, C. (1999). *Musique, geste, technologie*. Éditions Parenthèses.

Giannakis, K. (2006). A comparative evaluation of auditory-visual mappings for sound visualisation. *Organised Sound*.

Hamilton, R. (2008). q3osc: or how i learned to stop worrying and love the game. In *Proceedings of the International Computer Music Association Conference*.

Jacquemin, C., Ajaj, R., Cahen, R., Ollivier, Y., and Schwarz, D. (2007). Plumage: Design d'une interface 3d pour le parcours d'échantillons sonores granularisés. In *Proceedings of the Conférence Francophone sur l'Interaction Homme-Machine(IHM'07)*.

Jordà, S. (2005). *Crafting musical computers for new musics' performance and improvisation*. PhD thesis, Universitat Pompeu Fabra.

Levin, G. (2000). *Painterly Interfaces for Audiovisual Performance*. PhD thesis.

Mäki-Patola, T., Laitinen, J., Kanerva, A., and Takala, T. (2005). Experiments with virtual reality instruments. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05), Vancouver, BC, Canada*.

McCormick, C. Ergates: http://mccormick.cx/projects/ergates/.

Moody, N. (2006). Motion as the connection between audio and visuals.

Mulder, A. G. (1998). *Design of virtual three-dimensional instruments for sound control*. PhD thesis.

Naef, M. and Collicot, D. (2006). A vr interface for collaborative 3d audio performance. In *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06), Paris, France*.

Olive, J. and Pickles, S. Fijuu: http://www.fijuu.com/.

Olive, J. and Pickles, S. q3apd: http://selectparks.net/ julian/q3apd/.

Poupyrev, I., Weghorst, S., Billinghurst, M., and Ichikawa, T. (1998). *Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques*.

Rodet, X., Gosselin, F., Mobuchon, P., Lambert, J.-P., Cahen, R., Gaudy, T., and Guedy, F. (2005). Study of haptic and visual interaction for sound and music control in the phase project. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05), Vancouver, BC, Canada*.

Valbom, L. and Marcos, A. (2005). Wave: Sound and music in an immersive environment. *Computer and Graphics*.

Wakefield, G., Morin, J. K., Novak, M., Overholt, D., Putnam, L., Thompson, J., and Smith, W. (2008). The allobrain: an interactive stereographic, 3d audio immersive environment. In *CHI Conference Workshop on Sonic Interaction Design*.

Wozniewski, M., Settel, Z., and Cooperstock, J. (2006). A spatial interface for audio and music production. In *Proceedings of the International Conference on Digital Audio Effects (DAFx), 2006*.

Two approaches for HRTF interpolation

Gustavo H. M. de Sousa[1*], **Marcelo Queiroz**[1†]

[1]Computer Science Department
University of São Paulo
R. do Matão 1010
05508-900 São Paulo, SP

gustavo@ime.usp.br, mqz@ime.usp.br

Abstract. *This paper deals with the problem of interpolating HRTF filters for the binaural simulation of continuously moving sound sources. Two novel approaches are presented, one based on interpolating impulse responses in time-domain, and another based on interpolating poles and zeros of low-order IIR approximations of measured HRTF filters. Computational tests show that the proposed interpolation schemes are feasible for processing anechoic signals with varying directional information without audible discontinuities (clicks). We present an objective comparison of relative errors of interpolated HRTFs with respect to measured ones, and discuss some of the difficulties of evaluating the quality of the results from both objective and subjective points-of-view.*

Resumo. *Este artigo lida com o problema de interpolar filtros HRTF para a simulação binaural de fontes sonoras em movimento. Duas abordagens inovadoras são apresentadas, uma baseada na interpolação de respostas impulsivas no domínio do tempo, e outra baseada na interpolação de pólos e zeros de aproximações IIR de baixa ordem dos filtros HRTF. Testes computacionais mostram que os esquemas de interpolação propostos são viáveis para o processamento de sinais de áudio anecóicos com informação direcional variável sem introduzir descontinuidades audíveis (clicks). É apresentada uma comparação objetiva dos erros relativos das HRTFs interpoladas com respeito àquelas medidas, e são discutidas algumas das dificuldades em avaliar a qualidade dos resultados dos pontos de vista objetivo e subjetivo.*

1. Introduction

Research in sound spatialization is becoming increasingly important since the second half of the 20th century, from both technological and artistic points-of-view [Roads, 1996]. Its applications include immersive systems and virtual reality, computer games, and musical composition. In this latter field of application sound spatialization becomes an added dimension of artistic experimentation [Stockhausen, 1961, Xenakis, 1992].

A very simple and widespread multichannel sound spatialization technique is called *amplitude panning*, which creates a soundscape by changing the relative amplitude of the signals of each channel [Moore, 1990]. A straightforward extension of this technique is Vector-Based Amplitude Panning or VBAP, which allows for 3D loudspeaker configurations [Pulkki, 2001]. These techniques are intended for loudspeaker reproduction of a sound field; they pressuppose that the listener is seating at the center of the loudspeaker configuration space, and they have very limited results outside this privileged spot (or *hot spot*).

*Supported by CAPES and FAPESP.
†Supported by CNPq.

One alternative to sound field recreation through loudspeakers is the use of binaural simulation through headphones. Much more control can thus be gained over the subjective impression of spatial location of sound sources, and more flexibility on the number of simultaneous users in various different locations, each receiving its own individual soundscape, at the expenses of having to wear a pair of headphones and carry some wireless receiver. Nevertheless, it is a promising path for sound spatialization that imposes no budget or setup burdens.

Although simple schemes like amplitude panning and *Interaural Time Differences* (ITD) have been applied to binaural spatialization, much more realistic results may be obtained through the so-called HRTF filters (*Head-Related Transfer Functions*), which are usually obtained by directly measuring the effect of an incoming signal inside one person's ears (using tiny contact microphones), or by using a dummy head with internal microphones [Algazi et al., 2001]. These filters capture all effects of sound propagation from the original sound source to the listener's ears that modify an incoming signal, including interaural level and phase differences, absorption, reflections and refractions on the environment and around the body of the listener. Among these effects, those resulting from the reflections and refraction of sound on the torso, head and pinnae have a profound impact on the subjective impression of direction [Blauert, 1997, Hofman et al., 1998].

Each HRTF filter (or HRTF for short) is by definition tied to a particular direction of incoming sound, as specified during its measurement. It would be therefore necessary to have one measured HRTF for each direction of incoming sound one would like to represent in a simulation. Several databases of recorded HRTFs exist, such as the CIPIC database [Algazi et al., 2001], that represent large, although finite sets of incoming directions that may be readily used in auralization of static sound sources.

As opposed to that, we are interested in allowing virtual sound sources to describe arbitrary spatial trajectories, that should be subjectively perceived through headphones. In order to change smoothly from one HRTF to another, it is necessary to have interpolation schemes for these filters that satisfy two conditions: on one hand, no audible discontinuities (clicks) should be heared during transitions, and on the other hand, interpolated HRTFs should match as closely as possible the corresponding interpolated directions.

It should be noted that although this problem might seem an easy one at first glance, it remains unsolved in PureData and was only recently addressed in Csound, two very widespread and massively used sound processing computational environments; both Csound's opcode *hrtfer* and Puredata external *earplug~* produce clicks when a pure sinusoidal signal is made to wander around the listener's head; a newer Csound opcode *hrtfmove* uses crossfades to eliminate artifacts when switching between HRTF filters.

One interpolation scheme for HRTF filters, known as bilinear interpolation [Savioja et al., 1999, Freeland et al., 2002], aims at expanding the database of measured HRTF by constructing HRTFs for intermediate positions, so-called IPTFs or *Inter-Positional Transfer Functions*, from the available measured HRTFs. These are computed as linear combinations of four adjacent HRTF filters corresponding to a square around the desired direction (we use the term adjacent filters in the sense of filters corresponding to adjacent directions in a finite HRTF database).

We propose two new approaches for HRTF interpolation. The *triangular interpolation* is a small improvement over bilinear interpolation; it combines linearly three HRTFs corresponding to a triangle around the desired direction, with a 25% computational gain for obtaining each interpolated HRTF. The *spectral interpolation*, on the other hand, differs fundamentally from previous approaches in the sense that the HRTF database is substituted by a database of low-order IIR filters that approximate the original HRTFs, computed prior to simulation. Each filter is represented by a set of 6 poles and 6 zeros in the complex plane, that directly affect regions of resonance and antiresonance in the frequency response. By interpolating the positions of corresponding poles and zeros in adjacent filters, we construct intermediate filters that are not linear combinations of adjacent ones, but rather nonlinear combinations that are inherently tied to the spectral shapers (i.e. poles and zeros) of an IIR filter.

Section 2 presents the details of VBAP and the derived triangular interpolation. Section 3 presents the Kalman filter method for obtaining low-order IIR aproximations for the original HRTFs, a technique for matching poles and zeros of adjacent IIR representations, and the spectral interpolation technique for moving sound sources. The implementation of these methods, as well as computational experiments and their discussion are presented in section 4, and some conclusions and further work are presented in section 5.

2. Triangular Interpolation

The original motivation for the triangular interpolation technique was a transposition of the VBAP technique from its original context (amplitude panning over loudspeakers) to a binaural application, where each HRTF-auralized signal would be treated as one of the loudspeakers in order to compute the linear combination coefficients. We will first present the original VBAP technique in order to introduce formally the triangular interpolation.

2.1. Vector-Based Amplitude Panning

The VBAP technique, as mentioned earlier, is a special case of amplitude panning. It aims at recreating the subjective impression of (i.e. positioning) a virtual sound source by sending the same signal over several fixed loudspeakers, each with a different amplitude gain. If $x(t)$ is the original signal, loudspeaker n will play $x_n(t) = g_n \cdot x(t)$, for $n = 1, \ldots, N$, where g_n is the corresponding gain [Pulkki, 2001, Moore, 1990]. These gains depend on the position of each loudspeaker and of the virtual source.

In a tridimensional setting it is customary to consider sets of 3 adjacent loudspeakers forming a triangular cone centered on the listener, that are used for simulating virtual sound sources within the triangle. Let $\boldsymbol{l^n}, \boldsymbol{l^m}$ and $\boldsymbol{l^k}$ be the vectors corresponding to the direction of each loudspeaker relative to the listener, and $\boldsymbol{p}$ the direction of the virtual sound source. Then the gains g_n, g_m and g_k of the loudspeaker must satisfy $\boldsymbol{p} = g_n\boldsymbol{l^n} + g_m\boldsymbol{l^m} + g_k\boldsymbol{l^k}$ or, in matrix notation:

$$\boldsymbol{p}^t = \boldsymbol{g}\boldsymbol{L}_{nmk},$$

where $\boldsymbol{g} = [g_n g_m g_k]$ and $\boldsymbol{L}_{nmk} = [\boldsymbol{l^n}\boldsymbol{l^m}\boldsymbol{l^k}]^t$. The solution is therefore

$$\boldsymbol{g} = \boldsymbol{p}^t\boldsymbol{L}_{nmk}^{-1} = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} l_1^n & l_2^n & l_3^n \\ l_1^m & l_2^m & l_3^m \\ l_1^k & l_2^k & l_3^k \end{bmatrix}^{-1}.$$

The VBAP technique states that the subjective impression of a virtual sound source coming from diretion $\boldsymbol{p}$ is recreated by applying gains g_n, g_m and g_k to the loudspeakers, according to the above equation.

2.2. Using VBAP gains to interpolate HRTFs

Given an input signal $x(t)$, and a measured HRTF corresponding to a direction $\boldsymbol{l}$, given by a finite impulse response $h(n)$, the spatialized signal is obtained by the convolution

$(x*h)(t)$ [Moore, 1990]. The triangular interpolation scheme correponds to treating each spatialized signal $(x*h)(t)$ as coming from a fixed loudspeaker placed at $\boldsymbol{l}$, and applying the VBAP gains to those signals before mixing them in the binaural simulation.

If $\boldsymbol{l^n}, \boldsymbol{l^m}$ and $\boldsymbol{l^k}$ are the vectors corresponding to the direction of each measured HRTF, and $\boldsymbol{p}$ is the intended direction of the virtual sound source, then the gains given by $\boldsymbol{g} = \boldsymbol{p}^t \boldsymbol{L}_{nmk}^{-1}$ will be applied to the signals $(x*h_n)(t)$, $(x*h_m)(t)$ and $(x*h_k)(t)$, leading to an interpolated signal

$$y(t) = g_n \cdot (x*h_n)(t) + g_m \cdot (x*h_m)(t) + g_k \cdot (x*h_k)(t).$$

This is supposed to be done independently for each ear in a binaural simulation (each HRTF actually corresponds to a given direction and one of the ears of the listener).

By using the linearity property of convolution, it is easy to see that

$$\begin{aligned} y(t) &= g_n \cdot (x*h_n)(t) + g_m \cdot (x*h_m)(t) + g_k \cdot (x*h_k)(t) \\ &= (x*[g_n \cdot h_n + g_m \cdot h_m + g_k \cdot h_k])(t), \end{aligned}$$

which shows that the above interpretation is equivalent to combining the HRTFs directly with the VBAP gains, obtaining the interpolated HRTF

$$\hat{h} = g_n \cdot h_n + g_m \cdot h_m + g_k \cdot h_k,$$

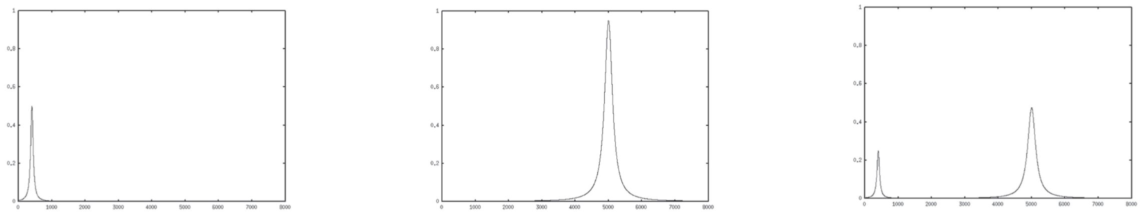
and then applying the convolution. The latter strategy corresponds closely to bilinear interpolation [Savioja et al., 1999], but using 3 adjacent HRTFs instead of 4. The triangular approach cuts down 25% of the computational cost of evaluating $y(t)$, regardless of which order of computation is chosen. For direct implementation, the first equation saves about M multiplications, where M is the size of the impulse responses; by using fast convolution on blocks of M samples, the second equation performs better, by requiring a single fast convolution (cost $M \log M$) instead of three.

This method may be applied on a sample-by-sample basis, meaning that the gains applied on each convoluted signal are updated at the audio rate. Since smoothly moving sound sources are bound to update their directions at a much slower rate, all gains will also behave smoothly, and that guarantees that no audible discontinuities will be introduced by the interpolation method. If one decides to update the gains at a slower control rate, this will not be a problem as long as the speed of the virtual sound source does not exceed a certain threshold, related to the rate of filter switching entering the audible range. One solution to the slow update implementation is to lowpass-filter the control signal that represent the directions of the moving sound source, thus forcing the increments of azimuth and elevation to be small compared to the control rate.

The triangular interpolation technique is essentially a time-domain technique, in the sense that the waveforms of impulse responses are combined and time-domain convolution is computed. Due to the linearity of the Fourier Transform, it is possible to apply the same interpolation procedure to the complex transfer functions representing the HRTF filters, using the same gains computed by the VBAP method, and substituting convolution by multiplication of spectra. This is an alternative implementation that does not alter the method at a conceptual level. In section 4 we will discuss important implementation details, such as dealing with phase differences of the HRTFs, and the effects of these in the comparison of interpolated HRTFs with measured ones.

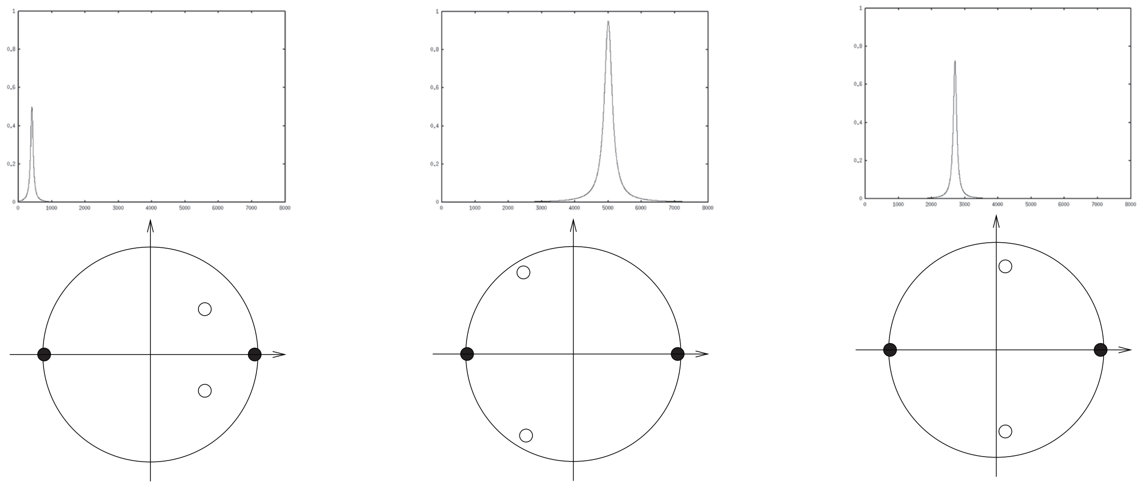
3. Spectral Interpolation

In the previous section we mentioned that the triangular interpolation may be implemented either in time-domain or frequency-domain, by applying linear gains to the impulse responses or to the transfer functions representing the HRTFs. In both cases we



Filter A | Filter B | Interpolated filter

Figure 1: two bandpass filters and a linear interpolation.



Filter A | Filter B | Interpolated filter

Figure 2: pole-zero diagrams and spectral interpolated filter.

were dealing with *linear combinations* of waveforms, impulse responses, or spectra. The spectral interpolation technique presented here aims at something fundamentally different, namely shaping the spectra of interpolated HRTFs in such a way that the regions of resonance and antiresonance are also "interpolated" (in a weaker sense, to be made precise in the sequel).

To illustrate the idea of spectral interpolation, consider a family of bandpass filters, defined by center frequency and bandwidth. We might consider a smooth transition from filter A to filter B in figure 1. With linear interpolation (of impulse responses, or equivalently of spectra) the intermediate filters wouldn't look like a bandpass filter at all (see the third plot in figure 1). By representing these filters as 2-poles-2-zeros IIR filters, it is natural to define an intermediate filter by placing the poles at an intermediate frequency (angle in the polar representation), and an intermediate distance from the origin (magnitude in the polar representation). This corresponds exactly to interpolating the polar representations of the poles in the pole-zero diagram (see figure 2).

IIR filters are well-known for their compactness: with few feedback coefficients they are able to encode very complex filter responses, thus making them much more efficient than FIR filters in terms of processing time. For spatialization purposes, this means being able to use more spatialization units in real-time and in parallel.

In order to extend this idea to HRTFs we first need to discuss how to obtain pole-zero representations for measured HRTFs. Then we need to address the problem of identifying corresponding pairs of poles or zeros in adjacent IIR filters. Finally we discuss the interpolation of poles and zeros and the use of intermediate IIR filters in the simulation of moving sound sources.

3.1. Obtaining low-order IIR filters

HRTF filters are usually found in the form of HRIR or Head-Related Impulse Responses, which are time-domain signals with a duration of a fraction of a second; the CIPIC database [Algazi et al., 2001], for instance, features recordings consisting of 200 samples on a 44.1kHz sampling rate. Although really small, these samples may be viewed as coefficients of a 200-zeros FIR filter, which is a lot of information in the pole-zero complex plane to deal with. In order to be able to manipulate the positions of poles and zeros we are bound to sacrifice precision in the representation of the filter and use low-order approximations (i.e. with fewer zeros and/or poles).

The Kalman method [Kalman, 1958, Kulkarni and Colburn, 2004] is designed to achieve optimal IIR approximations for a given FIR filter, in the sense that it minimizes the squared error of the approximation over all possible IIR filters of the same order. Suppose we want to approximate a given HRIR $y(n)$ using an IIR filter with P poles and Q zeros, corresponding to the filter equation

$$\hat{y}(n) = \sum_{i=0}^{Q} a_i x(n-i) - \sum_{j=1}^{P} b_j \hat{y}(n-j).$$

The Kalman method minimizes the squared norm of the approximation error

$$\sum_{k=0}^{M} e(k)^2,$$

where $e(k) = \hat{y}(k) - y(k) = \sum_{i=0}^{Q} a_i x(k-i) - \sum_{j=1}^{P} b_j \hat{y}(k-j) - y(k)$. This unconstrained convex quadratic optimization problem has a closed-form solution that is given by

$$\begin{bmatrix} a_o \\ \vdots \\ a_Q \\ -b_1 \\ \vdots \\ -b_P \end{bmatrix} = W^{-1} \left(\sum_{k=0}^{M} y(k) w(k) \right),$$

where $w(k) = (x(k), \ldots, x(k-Q), y(k-1), \ldots, y(k-P))^t$, for $k = 0, \ldots, M$, and the matrix W is a sum of the outer products $W = \sum_{k=0}^{M} w(k) w(k)^t$.

With this method and by choosing a small value for P and Q we are able to convert a complete database of FIR HRTF filters into low-order IIR filters that can be used in the spectral interpolation method.

3.2. Matching poles and zeros

One of the difficulties that appear as we try to interpolate pole-zero diagrams is to assign matching pairs of corresponding poles and corresponding zeros for each group of adjacent IIR filters. On one hand, typical databases such as the CIPIC are not dense enough to ensure that the Kalman method will produce sufficiently close diagrams for adjacent filters. This means that it is not always obvious how to transform one pole-zero diagram into another because we don't always know which pole in the first diagram goes to which pole in the second diagram (see figure 3 for an example). On the other hand, by making careless assignments of pole pairs we risk losing stability of the intermediate filters, as well as producing sound artifacts due to rapidly changing frequency responses.

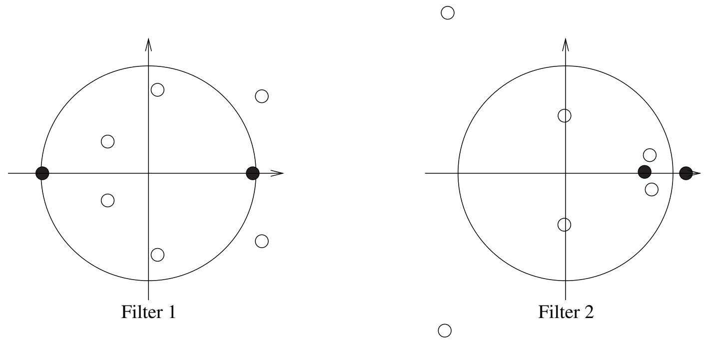


Figure 3: Two adjacent filters represented by pole-zero diagrams.

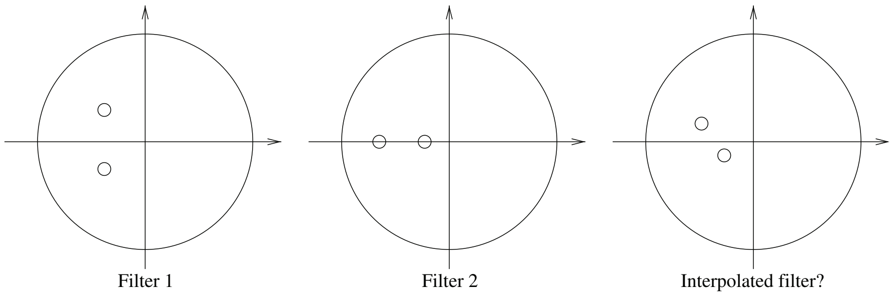


Figure 4: Interpolating from complex towards real poles.

In the absence of a reliable perceptual guideline for assigning such pairs, we adopted a risk-averse strategy, that corresponds to a minimum energy or least effort solution: to select the matching of poles and zeros that minimizes the overall motion between corresponding pairs. This is easy to compute for low-order filters; it suffices to consider all possible permutations of the first set of poles and zeros and evaluating the total distance to the corresponding poles and zeros of the second set. By remembering that all complex poles and zeros come in complex-conjugate pairs, the actual size of the sets where the permutation is applied may be halved.

A second difficulty in this matching is the treatment of poles and zeros that lie on the real line. These are actually the most difficult to deal with, because one cannot safely interpolate between a complex pole and a real one without destroying the complex-conjugate property of complex poles (see figure 4). Forcing the interpolated filter to assume such configurations would produce a filter equation with complex coefficients, which delivers a complex time-domain signal. The only way out is to treat complex poles and real poles as two separate categories (as we were treating poles and zeros as separate categories), and assigning complex poles in one filter to complex poles in the other (and likewise with real poles).

What this solution implies is that we must determine beforehand not only the number of poles and zeros, but also the number of real poles and the number of real zeros in the pole-zero representation. This imposes a constraint on the creation of IIR filters that the Kalman method alone is not able to enforce. One way to cope with this difficulty is to choose the more frequent pole-zero structure in the IIR filters produced by the Kalman method, and then adjust all non-conforming filters to this structure. This involves replacing pairs of real poles for complex poles and vice-versa, depending on the specific case. Substitutions such as those may be implemented by direct search of conforming filter candidates in the neighborhood of the problematic filter.

3.3. Using the interpolated filters

By using Kalman filter approximations to the original HRTF databases, and having a reasonable matching of poles and zeros for adjacent filters, we are now able to discuss the use of the spectral interpolation method for the spatialization of moving sound sources.

Suppose we want to simulate a direction $\boldsymbol{p}$, that lies in a triangular region of the HRTF database defined by the directions l_n, l_m and l_k, and let g_n, g_m and g_k be the gains computed by the VBAP method (section 2). Let p_l^j and q_l^j be the j-th pole and j-th zero of the HRTF $l \in \{n, m, k\}$, and suppose that their polar representations are $p_l^j = \alpha_l^j e^{i\varphi_l^j}$ and $q_l^j = \beta_l^j e^{i\omega_l^j}$. We define the j-th pole and j-th zero of the interpolated IIR filter as

$$\hat{p}^j = (g_n\alpha_n^j + g_m\alpha_m^j + g_k\alpha_k^j)e^{i(g_n\varphi_n^j + g_m\varphi_m^j + g_k\varphi_k^j)}$$

$$\hat{q}^j = (g_n\beta_n^j + g_m\beta_m^j + g_k\beta_k^j)e^{i(g_n\omega_n^j + g_m\omega_m^j + g_k\omega_k^j)}.$$

Each IIR filter also has an overall gain coefficient a_0; we accordingly define the interpolated coefficient as $\hat{a}_0 = g_n a_{0,n} + g_m a_{0,m} + g_k a_{0,k}$.

This set of poles and zeros define a transfer function given by

$$\hat{H}(z) = \hat{a}_0 \frac{\prod_{j=1}^{Q}(1 - \hat{q}^j z^{-1})}{\prod_{j=1}^{P}(1 - \hat{p}^j z^{-1})},$$

which may be rewritten as

$$\hat{H}(z) = \frac{\hat{a}_0 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \cdots + \hat{a}_Q z^{-Q}}{1 + \hat{b}_1 z^{-1} + \hat{b}_2 z^{-2} + \cdots + \hat{b}_P z^{-P}},$$

from which the coefficients for the filter equation

$$\hat{y}(n) = \sum_{i=0}^{Q} \hat{a}_j x(n-j) - \sum_{j=1}^{P} \hat{b}_j \hat{y}(n-j)$$

are readily available.

This method may be likewise implemented in a sample-by-sample update basis, and so the same comment of section 2.2 regarding absence of audible discontinuities apply. If the virtual direction varies smoothly, so do the VBAP gains and therefore all poles and zeros also vary smoothly, and since the filter equation coefficients are continuous functions of the poles and zeros, the output signal will not be affected by audible discontinuities. If one so desires, a slower control rate for updating the filter equation may be employed, provided that the motion of the virtual sound source is slow (or lowpass-filtered) compared to the filter switching rate.

4. Implementation and Discussion

We will discuss in the sequel some implementation details for each of the proposed methods, and also some computational experiments that we made to assess the quality of the interpolations.

4.1. Implementation of the triangular interpolation and discussion

One of the issues that appeared in the implementation of the triangular interpolation method had to do with phase differences between adjacent HRTF filters. This is a sneaky and yet important detail, because it severely affects the interpolated filters.

We used mainly the CIPIC database [Algazi et al., 2001] as input for our experiments. In those recordings, the initial silent gap does not correspond to the distance from the sound source to the ear of the dummy head, but rather it is freely adjusted in order to preserve the main characteristics of each HRIR waveform. That means that adjacent HRIR may have quite different onset times of the direct sound, and thus cannot be carelessly combined. We alleviate that problem by synchronizing onset times of adjacent HRIRs that were going to be used in a triangular interpolation, applying gains and mixing, and only then adjusting the overall delay based on a geometric model of the CIPIC recording settings.

In order to try to objectively evaluate the quality of this interpolation scheme, we considered an automatic experiment where original, measured HRIRs corresponding to given directions would be compared to interpolated versions, for the same directions, obtained from adjacent HRIRs. For each possible HRTF in the database, corresponding to a certain azimuth and elevation angles, we tried to simulate this HRTF by interpolating HRTFs in adjacent azimuth and elevation angles. More specifically, if D is point in the database, we chose triangles ABC containing D, such that A, B, and C are direct neighbors of D. Here all HRTFs for positions A, B, C and D are known, and an interpolated version D' is computed from A, B, C and the corresponding VBAP gains. We then plot a graph of the magnitude of the frequency responses of D and D' (measured in dB), for selected frequencies, as a function of azimuth and elevation angles (see figure 5).

By comparing each pair of graphs in figure 5 one sees that the attenuation patterns of the original HRTFs (left column) are roughly preserved in the interpolated HRTFs (right column). The above frequencies were chosen for illustration purposes only, but as a rule the same preservation of patterns is observed for all frequencies. This observation supports the claim that the interpolation scheme preserves frequency-dependent directional information of the HRTFs.

We have also compared the same pairs of original and interpolated HRTFs by considering the relative errors given by $\frac{\|HRTF(D)-HRTF(D')\|}{\|HRTF(D)\|}$. This measure subsumes attenuation differences for all frequency components at once, but it does not behave very smoothly over the whole database. This is explained by the fact that the original HRTFs also do not behave smoothly on the whole range of possible azimuth and elevation angles, i.e. they vary more rapidly on some regions, and this might explain the difficulty of the interpolation scheme to produce close results in the sense of obtaining similar waveforms by using neighboring HRTFs.

An interesting issue is the relation of the relative errors with the size of the triangles used in the above experiment. We compared the errors of the above experiment with a new setting were points A, B, and C surrounding D were allowed to be neighbors of the neighbors of D (they had distance 2 in terms of database points). It came as no surprise that the errors in the new setting were always larger than the ones in the original experiment. This might indicate that the actual errors of using the triangular interpolation within the small triangles defined by the points in the CIPIC database is smaller than what we have obtained in the numerical experiment. This observation underlines the importance of the density of the database as expressed by the number of directions included in the measurements: the more thinly spaced these measurements are made, the better the resulting spatialization of a moving sound source.

4.2. Implementation of the spectral interpolation and discussion

As discussed in section 3, having a database of low-order IIR aproximations of the original HRTF database is a prerequisite for applying the spectral interpolation. We therefore applied the Kalman method on the CIPIC database, obtaining IIR filters of 6 poles and

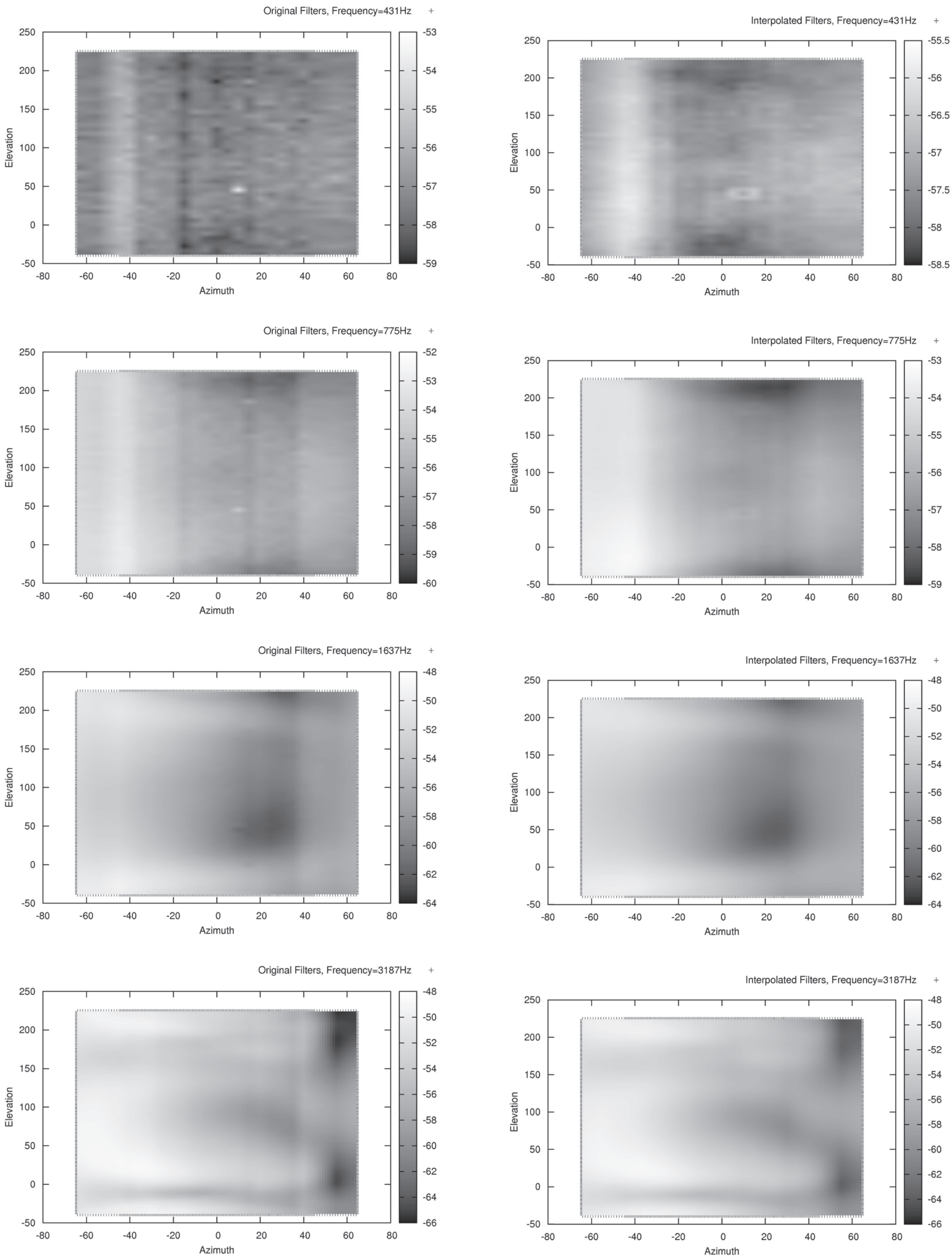


Figure 5: Comparison between original and interpolated HRTFs for selected frequencies. Each graph shows the pattern of attenuation for a given frequency as a function of the direction of incoming sound.

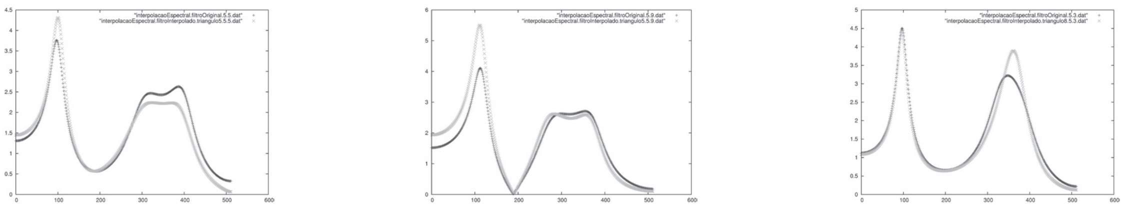


Figure 6: Measured and interpolated HRTFs.

6 zeros each. The first difficulty here was the structure of real and complex poles, as discussed in section 3.2. Since most of the Kalman filters had a similar structure of 4 complex poles and 2 real poles, we forced all non-conforming filters into this common structure.

With respect to an objective evaluation of the quality of the interpolation, it must be made clear that there are two different kinds of errors involved. First, there are the errors produced by approximating the original 200-point FIR filters for 6-pole-6-zeros IIR filters using the Kalman method. These are a price to be paid on the accuracy of the simulation for bringing the computational cost down to a much lower level, which supposedly have benefits on reducing real-time and parallelization constraints. Second, there are the errors introduced by using interpolated versions of the IIR filters. These are not to be compared with the original FIR filters (otherwise we would count the first kind of error twice), but with the IIR filters obtained by the Kalman method. In terms of the experiment of section 4.1, we consider the IIR filter corresponding to directions A, B, C and D to have been obtained with the Kalman method, and compare the IIR at D with the interpolated version of IIR at D obtained by using the VBAP gains on the IIRs of A, B and C. Figure 6 shows graphs of measured and interpolated HRTFs at selected directions of the database.

The method of spectral interpolation proposed here might be enhanced in several and important ways, for instance, by using HRTF databases more complete than CIPIC, and also by refining the pole-zero model using higher-order filters, thus allowing the simplified IIR model to capture more complex phenomena such as shoulder or pinnae reflections.

5. Conclusions and Further Work

In this article we presented two novel approaches for interpolating HRTF filters in the simulation of moving sound sources. The triangular interpolation is a time-domain linear interpolation technique akin to the bilinear interpolation, and the spectral interpolation presents a whole new way of dealing with moving sound sources.

With respect to allowing smooth transitions between filters for continuously moving sound sources, both methods may be made to update the interpolated filters on a sample-by-sample basis, which guarantees a smoothly varying output signal without audible discontinuities. Other strategies, such as lowpass-filtering the motion signal, may be used to ensure smooth transitions even when using a slower control rate.

One of the (already expected) conclusions of the experiment with the triangular interpolation is that the density of the HRTF database has a profound impact on every interpolation scheme. As the number of available measured HRTFs increases, the smaller the distance of a virtual sound source to an original HRTF will be, and the smaller the corresponding interpolation error.

We presented an attempt at objectively evaluating the quality of both methods, by recreating interpolated filters on top of existing ones and measuring the relative errors. The preservation of the overall attenuation patterns for all frequencies endorses the interpretation that frequency-dependent directional information is also preserved. The relative errors of interpolated filters were also measured, but these values were not well-behaved and did not help in providing objective conclusions about the quality of the interpolation procedure. This method of objectively comparing original and interpolated HRTFs proved to have a limited power in assessing the usefulness of these methods in the context of practical spatialization, where the illusion of direction is actually more important than a rigorous physical evidence of the relation between the direction and the corresponding filter.

Subjective tests are usually difficult to apply and to analyse for a number of reasons, such as our perceptual equipment lack of accuracy in differentiating nearby directions, and the fact that we are highly suggestible to visual imagery and other hints when evaluating direction of incoming sounds. Nonetheless, subjective tests are strongly needed in order to provide the clues to the relation between nearby directions and nearby HRIR waveforms or nearby HRTF frequency responses, and also to reconfirm the psychoacoustical effectiveness of these methods in simulating moving sound sources.

References

Algazi, R., Duda, R. O., Thompson, D. M., and Avendano, C. (2001). The CIPIC HRTF database. In *WASSAP '01*. 2001 IEEE ASSP Workshop on Application of Signal Processing to Audio and Acoustics.

Blauert, J. (1997). *Spatial hearing: the psychophysics of human sound localization*. MIT press.

Freeland, F. P., Biscainho, L. W. P., and Diniz, P. S. R. (2002). Efficient HRTF interpolation in 3D moving sound. In *WASSAP '01*. AES 22nd International Conference: Virtual, Synthetic, and Entertainment Audio.

Hofman, P., Van Riswick, J., and Van Opstal, A. (1998). Relearning sound localization with new ears. *Nat Neurosci*, 1(5):417–421.

Kalman, R. E. (1958). Design of a self-optimizing control system. In *Transactions of ASME*, volume 80, pages 468–478.

Kulkarni, A. and Colburn, H. S. (2004). Infinite-impulse-response models of the head-related transfer function. *The Journal of the Acoustical Society of America*, 115(4):1714–1728.

Moore, F. (1990). *Elements of computer music*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

Pulkki, V. (2001). *Spatial sound generation and perception by amplitude panning techniques*. PhD thesis, Helsinki University of Technology. Espoo, Finland.

Roads, C. (1996). *The computer music tutorial*. MIT press.

Savioja, L., Huopaniemi, J., Lokki, T., and Väänänen, R. (1999). Creating interactive virtual acoustic environments. *J. Audio Eng. Soc.*, 47(9):675–705.

Stockhausen, K. (1961). *Two lectures*. T. Presser.

Xenakis, I. (1992). *Formalized music*. Pendragon Press Stuyvesant, NY.

Creating Evolutionary Soundscapes with Gestural Data

José Fornari[1], Mariana Shellard[2], Jônatas Manzolli[1,2]

[1]Núcleo Interdisciplinar de Comunicação Sonora – NICS

[2]Institute of Arts - IA

Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6166 – 13.083-872 – Campinas – SP – Brazil

{tutifornari,marianashellard}@gmail.com, jotamanzo@hotmail.com

Abstract. *In Generative Art produced with computational support, gestural data has been increasingly considered an important source of information that is, at the same time, intuitively created and conveying the artistic meaning of the resulting artwork. In the same way, the concept of self-organization has been used in the study of human creativity and art production. This article describes the implementation of two interactive multi-modal artworks that use these two principles to compute adaptive sonifications. As gestural data, the first implementation uses the mapping of a handmade drawing collection. The second one uses the retrieval of body action movements performed by a dancer. The resulting soundscapes are created by a dynamic system implemented in PD (PureData). It uses principles of Evolutionary Computation (EC), which yields to the creation of a synthetic adaptive population of sound objects based on the retrieved gestural data. That emulates the biological evolution of individuals undergoing the processes of selection and reproduction. The overall sound is constantly generated by all individuals within the population. This is the system output, which can be described as a self-organized synthetic soundscape engendered by the initial artistic generated gestural information.*

Keywords: generative art, gesture, evolutionary algorithm, soundscapes, sound synthesis.

1. Introduction

The literature describes that, throughout Western Art History, the focus was into the resulting artistic object as the final production of an artist endeavor. However, in the 1950s, probably due to the advances of electronic technology, the artistic process started to take over and slowly equaling with the materiality of the final product, bringing about new artistic ideas and concepts. Lucy Lippard, when analyzing the artistic production of Sol LeWitt, said that his work was based on the premise that its "concept or idea is more important than the final object" [Lippard, 1973]. This concept is similar to the one of Generative Art, which is defined as any form of art where "a system, with a set of defined rules and some degree of autonomy, is put on movement" [Galenter, 2003].

Nevertheless, generative processes was already explored in music, even before computers had flourished. Few centuries ago, around the 1650s, priest *Athanasius Kircher*, based on the belief that musical harmony reflected the proportions of the universe, wrote a book entitled: *Musurgia Universalis*, in which he described the design of a musical generating machine [Cramer, 2005]. In 1793, *Hummel* published a system to generate musical score, whose creation is attributed to *Mozart*. In this system, music was generated by a random process, based on a dices tossing game. This system embeds most of todays generative art elements, in which a musician can create, from simple building blocks (predefined musical bars), a myriad of original compositions. Later, this was known as the *Mozart's dice game* and influenced many composers, such as *John Cage* and *Hiller Lejaren* to create a musical piece entitled HPSCHD [Husarik, 1983].

In visual arts, for the processual artwork approach, the act of drawing can be seen as the registration of artistic gestures. This process is analyzed by *Walter Benjamin* as coming from "...another level within the human psyche. It is a locus for signs by which we meet the physical world" [Dexter, 2005]. This aspect can be compared to the technical principle of computer programming language, as defined by *Cramer*, when he commented that programming is similar to "controlling matter through the manipulation of symbols" [Cramer, 2005]. An artwork autonomously created by a generative process, such as a computational system, is not restricted within an specific field, but in a multitude of different areas of knowledge, even beyond visual arts and music. Adaptive methods, such as Artificial Intelligence, Neural Networks and Evolutionary Computation, can then be seen as technological strategies that fit the principles of generative artwork, due to their ability of creating an artistic process that is dynamic and immersive.

Here we describe two artwork implementations using gestural data and evolutionary computation methodology. The first one is an installation where digital images from handmade drawings beget dynamic soundscapes created by evolutionary populations of sounds. This generative process continuously produces new sonic material, based on graphical features retrieved from these images. The second one is based on dance gestures collected using accelerometers attached to the limbs of a dancer. These movements were represented as time series describing these body parts displacements. They were later translated into a population of sounds that altogether produces the soundscape.

Both data types (drawings and dance movements) share the common characteristic of being similar and variant, which means that all drawings from the collection, as well as all dance movements are akin but not identical. This resembles a biological population where individuals are similar, as they belong to the same specie, although there are no occurrence of clones. These data were mapped into what is named here as sonic genotypes, which represents the acoustic characteristics of a sound object, here taken as an individual belonging to a synthetic population. The following section describes the foundations of the evolutionary soundscape system. Section three describes the mapping from gestural data to the genotypes that feeds the evolutionary system. Section four presents the implementation of the evolutionary soundscape system into PD (PureData). In section five we conclude this work with a brief discussion of the achieved results and possibility of forthcoming implementations.

2. Evolutionary Soundscapes

A soundscape can be seen as the acoustic corresponding of a landscape; a sonic environment that, in spite of constantly presenting a stream of original acoustical information, never repeats itself, but has plenty of unique perceptual features in a way that, with only acoustical cues, our mind can easily recognize and set it apart from other soundscapes [Truax,2008]. We can hear examples of natural soundscapes in locations nearby waterfalls, inside a forest, during a traffic jam, in a crowded central station, and so forth.

It is interesting to note that Schafer formally described soundscapes as "natural, self-organized processes usually resultants of an immense quantity of sound sources, that may be correlated or not, but conveys an unique sonic experience that is at the same time recognizable and yet always original" [Schafer,1977]. This can be seen as an open complex system with self-organized emergent properties. About the attempt of artificially creating soundscapes, Truax noted that: "soundscape composition might aim to computationally emulate self-organized biological or natural acoustic environments" [Truax,1978], which is one of the goals of the work here presented. Still in [Schafer,1977], it is defined three types of sonic elements that compound a soundscape. They are: 1) keynotes, 2) signals and 3) soundmarks. Altogether, they weave the immersive sonic environment of a soundscape. Keynotes are the sonic elements that define a soundscape, although they may not always be present or consciously perceived. Signals are the foreground sonic elements, always present in the soundscape. Soundmarks are the sonic elements that are unique to one specific soundscape, which gives its identity and set it apart from other soundscapes.

Several methods for designing an artificial soundscapes were tried already, such as the ones described in [Blauert, 1997], [Pulkki, 1997] and [Chowning, 1970]. They are mostly based on the parametric control of sound-sources (i.e. sound localization cues, random appearances, etc.) but, when compared to natural soundscapes, they still lack the ability of creating a truly self-organized processes. In a systemic viewpoint, we named as self-organization the phenomenon presented by certain complex systems that are opened and formed by the interaction of a variant group of agents (for soundscapes, agents are sound-sources). The interaction of all internal and external agents are perceived by the mind as an emergent self-similar process. As an example, a biological population can be seen as a self-organized system, as it is complex, opened (dynamically exchanging individuals) and self-similar (with an identity). In this work we approach two important features of a natural soundscape: sonic location and self-organization.

There are many techniques that help to emulate the sonic localization field of a soundscape. Some of the most usual ones are: Interaural Time Difference (ITD) [Kelly,1991], Interaural Level Difference (ILD) [Birchfield, 2005] and Head-Related Transfer Functions (HRTF) [Brungart,1999]. ITD cues refer to the time difference for the acoustic waves coming from one single sound-source, to arrive in both ears of one listener. Similarly, ILDs describe the difference of its intensity arriving in both ears. HRTFs, however, are a collection of spatial cues, given by digital filters, that represents the sound processing of the listener's body anatomy, such as the head shape and size, outer ears and torso. ITDs and ILDs can be easily emulated by a computational model. ITDs can be assessed by the time-delay variation between audio channels and it delivers a convincing sound-source localization of its azimuth angle within the horizontal plane. It was used in studies such as in a robotic sound source localization system [Murray, 2004].

Regarding the self-organization of a soundscape; it is known that adaptive computing methodologies can produce emergent, self-similar complex systems [Holland, 1992], [Holland,1996]. Among those, there is the Evolutionary Computation (EC); a method that is inspired in the problem-solving approach observed in nature. This method seeks out, in evolutionary steps, for the best solution among a landscape of possible solutions. Since 2001, the researching group at NICS (Interdisciplinary Nucleus of Sound Communication) has worked with EC in sound design and music composition. Some of these techniques created highly textured sonic outputs, which is a feature found in natural soundscapes [Fels, 2001], [Manzolli,2001]. We developed the *ESSynth*; a system using EC principles, in which a population of waveforms evolves along time, in generation steps, by the action of genetic operators and a fitness functions, where the sound output is the overlapped queue of best-individuals of each generation [Manzolli, 2001], [Fornari, 2001]. Later, we incorporated sonic spatial localization cues in this method [Fornari, 2006], [Fornari, 2007]. These are based on the application of concepts from the theory of Complex Adaptive Systems (CAS) for sound synthesis [Caetano, 2007]. As described in [Holland,1992], CAS consists of a large number of agents with interconnected parameters that, altogether, exhibits coherent emergent properties. It is also known that CAS can generate emergent properties by means of its agents competition and/or cooperation [Holland,1996]. Its systemic behavior is the result of interactions between a large number of its formant agents, leading to the process of self-organization, in which a CAS may pass through several organizational states [Foerster, 1960]. We investigated here whether such process can also be created by an ESSynth model to generate soundscapes [Caetano, 2007]. Originally, ESSynth was based on a population of waveforms (the ESSynth "individuals"). Each individual had its own genotype; a group of psychoacoustic curves that defines how its waveform is perceived and understood.

In this work we aimed to apply an ESSynth method to generate emergent sonic properties based on external aspects. Our model of individual is an algorithm (a Pd patch) in which a genotype is described by few acoustic descriptors, initially mapped from the gestural data and simple interactive parameters, to control the evolutionary soundscape system. As it is later described, this seemed to suffices for the creation of dynamic soundscapes.

Here we implemented an evolutionary system with a variable-size population that starts from only two individuals. As the evolutionary process progresses, new individuals will be born and aging individuals will dye. The output is a soundscape generated by all active (alive) individuals within the population. Individuals will also move inside the population and its location will be perceived as a moving sound source. By proximity, individuals will generate offsprings whose genotype is produced by genetic operators, from the two genotypes of its parents. Each genotype is made of six arrays. Each array controls one acoustic parameters. They are organized in two main blocks: tonal and stochastic. The tonal part is in charge of tonal sounds (presenting pitch). It has three parameters of control: 1) intensity, 2) frequency and 3) distortion. The stochastic part is responsible for the generation of noisy, or percussive, sounds and has also three controlling parameters: 1) intensity, 2) center frequency of a band-pass filter, and 3) distortion, given by this filter bandwidth. It is interesting to notice that, with this implementation, the distortion parameters make a bridge from tonal to stochastic. Without distortion, tonal part generates a sine-wave sound and the stochastic part delivers white-noise. As the distortion rate increases, the tonal part output goes toward a noisy sound (made by the waveform clipping) and the stochastic part, by the constriction of its filter bandwidth, tends to become more tonal, a whisper-like sound.

By default, each array has a fixed length of 100 elements, real numbers, normalized between [-1,+1]. They are time series representing the independent variation of each acoustical parameter. Each initial genotype will receive a matrix M(6x100) from the previous mapping of gestural data. From top to bottom, the M matrix lines are respectively: tonal intensity, tonal frequency, tonal distortion, stochastic intensity, stochastic frequency, stochastic distortion.

3. Retrieving Gestural Data

Gesture, as an artistic expression, is here seen as the movements and actions embodying artistic intention. So, by retrieving its data, it may be possible to access information that resembles, at some extent, the art meaning contained in the final artistic objects (drawings) or action (dance), and expressing it in the sonic domain by using this data to generate dynamic soundscapes throughout an evolutionary system. The retrieval of such data is described in the next sub-sections.

3.1. Gestures from Drawings

We started with drawings picked from a large collection (over 200 conceptual drawings) that are very similar but, as they were hand made, never identical to each other. They were all created through the repetition of a similar, back-and-forth gesture. This naturally creates a collection that resembles a biological population of individuals belonging to the same specie, although, its correspondent evolution only occurred during its artistic process of creation. By using the ESSynth method, it seemed feasible to develop an artistic installation in which characteristics of each drawing could be mapped into genotypes of individuals (sound objects) gathered into a population where its continuing evolution creates a dynamic soundscape.

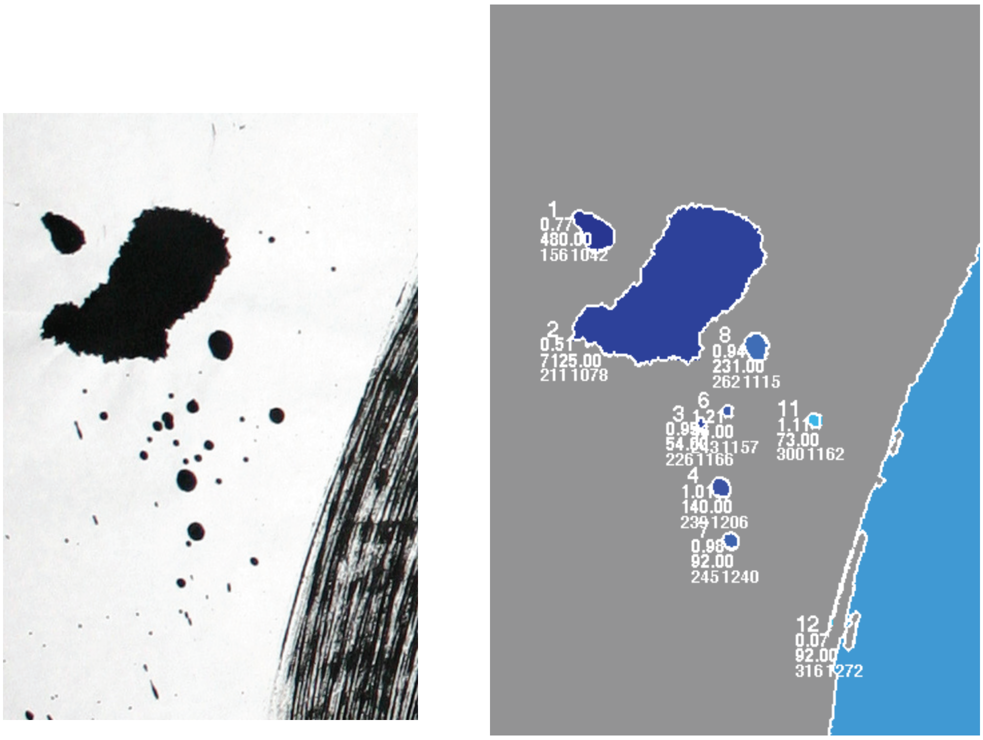


Figure 1.Mapping a conceptual drawing.

The first step was the creation of a method to map drawing features into sonic features. Figure 1 shows, on the left side, details of the digital image of an original drawing belonging to this collection. On the right side, it is shown the mapping of this image, done by an algorithm developed in Matlab. Note that the mapping collects several objects belonging to the same image. Each object has also several features associated with it. Some of them are shown in the Figure above, imprinted at the left side of each object.

By analyzing these objects, we considered that they belong to three types that, altogether characterize the "individuality" of each drawing. Such graphical elements are found in all drawings. They are here named as: Cumulation, Repetitions and Fragments. Cumulation is the biggest object found in one image. There is only one cumulation per drawing. It is usually given by the concentration of paint at the bottom of the image, where the drawing gesture initiated. Repetitions are objects with a stretched shape. They are normally the quasi-parallel traces found at the middle of the drawing, generated by the back-and-forth gesture. Fragments are small, detached and circular spots of paint dripped at the outlying parts of the drawing, spilled due to the gesture intensity. Following that, we related each graphical element with a single sonic aspects that seems to synesthetically represent, in the acoustic domain, each graphical aspect of the drawings. We related the object Cumulation to stochastic low-frequency sonic features, steady and with longer duration. Repetitions were related to tonal sounds, middle-range frequencies and middle time duration. Fragments were related to short time duration , like sparks of either stochastic or tonal sounds. Each image mapped generates several graphic objects, one is the cumulation and the others are either fragments or repetitions. Each object has several features associated with it. They are mapped into a matrix with the genotypes of the initial individuals that will start the evolutionary system.

We used the projection of the bi-dimensional shape of each object into horizontal and vertical coordinates to create the time series of 100 elements. For tonal sounds, we used its horizontal projection. For stochastic sounds, the vertical projection was used. The distortion rate was given by the difference between horizontal and vertical projections. Then, each projection was circularly shifted according to the distance between its object and the image origin. The tonal intensity curve is the blend of all horizontal projections modulated by each object eccentricity parameter. Tonal frequency curve is the blend of all vertical projections modulated by the respective objects normalized angles of orientation. Tonal distortion is the blend of the projections difference, modulated by the inversion of its eccentricity. The stochastic intensity curve is the blend of all horizontal projections modulated by the square of each object normalized area. The stochastic frequency is the blend of all vertical projections modulated by the respective objects normalized angle of orientation. Stochastic distortion is the blend of the projections difference modulated by its eccentricity.

3.2. Gestures from Dance Movements

Rudolf Laban, a famous choreographer and movement theoretician, in his work: the Laban Movement analysis [Pforsich, 1977], postulated eight types of Basic Movement that are the combination of three independent categories of Effort Actions (Space, Weight and Time). They are: Float, Punch, Glide, Slash, Dab, Wring, Flick, and Press. These actions have been used by several acting and dance schools as movements embodying specific emotions. These gestures can be retrieved and used in several ways. For instance, the InfoMus Lab, has developed the software EyesWeb, a multimodal interactive systems for the real-time analysis of movements and acquisition of expressive gesture [Mancini,2007]. Here at Unicamp, in a collaboration between NICS and the Interdisciplinary Group of Theater and Dance, a performance called *Elementaridades* was developed, inspired in the physical movement of particles of matter, and its application of Rudolf Laban´s principles of movement in dance [Maia et al, 2001].

In this work, similar gestures were collected as movement data, using as gestural interface two Wii remotes (Wiimote) and their accessory, the Nunchuck. Each part of these four units (2 Wiimotes and 2 Nunchucks) has embedded an accelerometer that transmit wirelessly, via bluetooth, the real-time acquisition of seven motion parameters. Three of them are named, in aviation terms, as: yaw, pitch and roll. They referred to the accelerometer rotation around each of its three spatial axes [LaValle, 2006]. The next four parameters transmitted are: x, y, z (for each axis rotation raw angle) and *accel* (raw acceleration movement, disregarding its direction).

The equations below show the rotation matrixes describing the correlation between: yaw, pitch and roll with its rotation about the orthogonal axes, related to each respective angle: x, y and z.

$$yaw(x)=\begin{pmatrix}\cos(x) & -\sin(x) & 0\\ \sin(x) & \cos(x) & 0\\ 0 & 0 & 1\end{pmatrix}\quad pitch(y)=\begin{pmatrix}\cos(y) & 0 & \sin(y)\\ 0 & 1 & 0\\ -\sin(y) & 0 & \cos(y)\end{pmatrix}\quad roll(z)=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos(z) & -\sin(z)\\ 0 & \sin(z) & \cos(z)\end{pmatrix} \tag{1}$$

The data was collected by a computer model (a PD patch) that recorded each movement in synchronism with the seven parameters of each one of the four accelerometers (given a total of 28 time-series) sampled at every 50 milliseconds. The accelerometers were attached at the dancer's knees and elbows. The resulting data was given as a text file, automatically created by this patch.

Table 1 shows the eight body actions described by Rudolf Laban and its formant aspects. The movements retrieved were performed by the a dancer according to the premisses shown in this Table.

Table 1. Body Actions, as described by Rudolf Laban

Action	Space	Weight	Time
Sliding	Direct	Light	Slow
Fluctuating	Flexible	Light	Slow
Punctuating	Direct	Light	Rapid
Shaking	Flexible	Light	Rapid
Pressing	Direct	Firm	Slow
Twisting	Flexible	Firm	Slow
Punching	Direct	Firm	Rapid
Whipping	Flexible	Firm	Rapid

These eight body movements were recorded and processed by the PD patch. The Figure 2 shows two random scenes, with image processing, to pinpoint the movement trajectories of each body actions recorded while the gestural data was collected in real-time by the wireless interfaces. The Figure 2, at the left side, shows a moment where the dancer was less active, almost standing still. The Figure 2, at the right side, depicts a moment where the dancer was very active, quickly waiving her arms, as they almost disappear from the image. It is possible to see, as a detail, particularly in the left image, two of the four accelerometers attached at the dancer's thighs and arms.

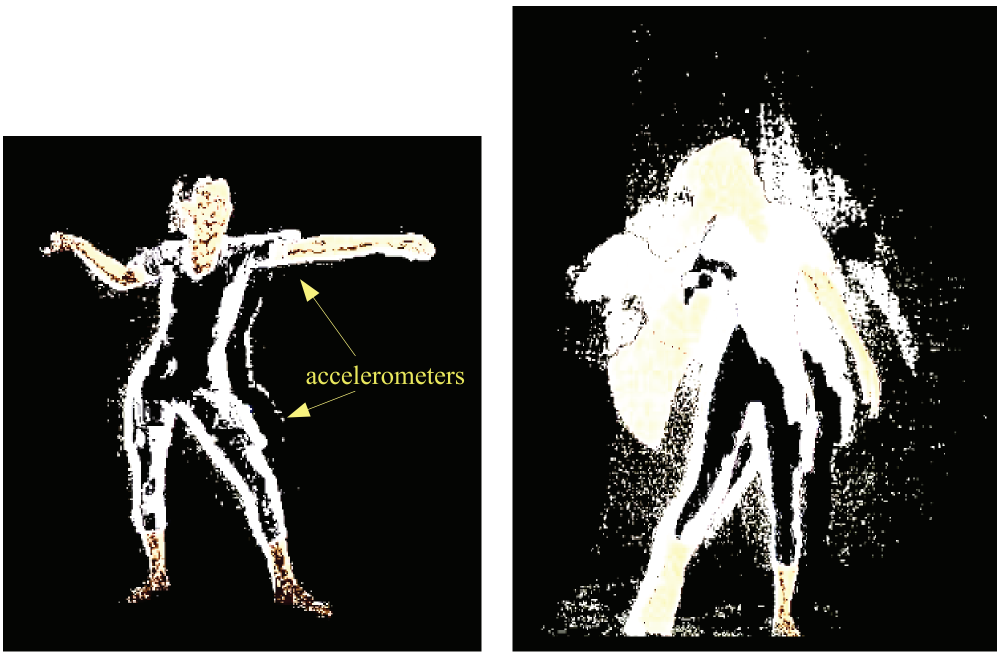


Figure 2. Scenes of the body actions recordings.

Using these interfaces, it was possible to collect gestural data out of the body movements, wirelessly and in real-time. Figure 3 shows a segment of the time series collected by one of the four accelerometers, where there are 7 synchronous curves: pitch, roll, yaw, acceleration, three raw rotation angles: x, y and z. They were then used to create the matrix M(6x100), corresponding to the individual genotypes that initiate the population, in the evolutionary system. These six arrays, compounding the genotype, were fed with data collected from one specific body action recording. In this implementation, we translated the mean variation of the accelerometer parameters attached in the dancer's arms to the tonal intensity and tonal frequency arrays. Similarity, the ones attached to her legs were translated to stochastic intensity and stochastic frequency. The tonal distortion was created from the difference between tonal intensity and frequency, as well as the stochastic distortion, given by the difference between stochastic intensity and stochastic frequency.

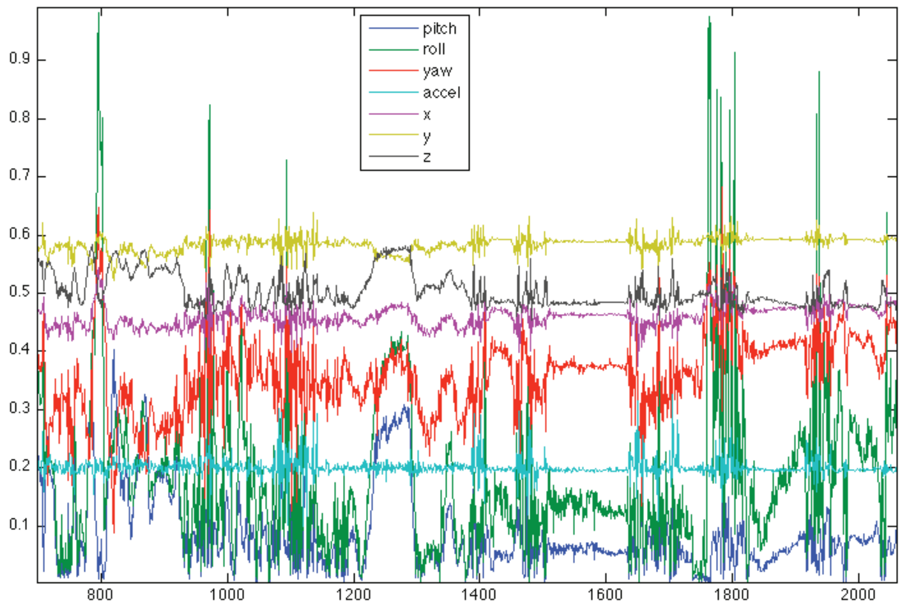


Figure 3. Segment of one body movement. This collected data was given by one accelerometer.

4. Implementation of the Evolutionary System

PD is an open-source visual programming language used for the implementation of real-time multimedia installations (www.puredata.org). A program developed in PD is called a patch, made of interconnected objects. They can be ether: preset objects, sub-patches (a patch inside the main patch) or abstractions (a separate patch that works as an object inside the main patch). An interesting feature of PD programming language is the ability of developing patches that are able to create and control other objects and patches. This follows the meta-programming paradigm, in which code can be written by code, without human intervention. There are recent efforts in the development of objects better shaped for meta-programming, such as the *iemguts* library, been developed by *Iohannes Zmölnig*, that aims to emulate self-aware agent system [Zmölnig, 2008]. Nevertheless, PD is already capable of exploring, at some extent, the automatic generation of patches by other patches.

The evolutionary sound synthesis, as originally introduced in [Manzolli, 2001] has a population of individuals, a Target set and two dynamic processes: 1) reproduction and 2) selection. The Target set guides the evolutionary process, similarly to the conditioning environmental pressure over a biological population. Selection uses fitness function to select the population individuals by measuring their fitness. It eliminates individuals not fit and find the fittest one, according to the sonic characteristics of individuals within the Target set. Reproduction uses the genetic operators: crossover and mutation, to create new individuals, offsprings of the best individual and the other ones within the fixed-size population.

In this work we used instead a variable-size population that starts with few individuals whose genotypes where mapped from a select group of gestural data. Then, the reproduction process creates new individuals. The output sound is given by the coexistence of all population individuals, which generates the dynamic soundscape. Figure 4 shows the implementation of the reproduction process. This figure depicts, on the top, twelve tables. On the left side, there are six tables. They belong to the genotype of the first individual. On the right side, there are the correspondent six tables with data from the genotype of a second individual. The genotypes are the mappings from the gestural data (drawings or movements) that are stored as text files. Both have the same organization. They are formed by seven lines, each one finished with a semicolon. The first line can be either the word "active" or "inactive", that informs the evolutionary system whether this individual is active, where active means that this individual is "alive" and can be picked by the reproduction process to create a new individual, as shown in Figure 4. The other six lines represent one of the six arrays of the individual genotype. Each line is a sequence of one hundred normalized real numbers (from -1 to +1). They all receive the data from the gestural matrix M(6x100).

Individuals are implemented as a PD abstraction (a separated patch). Each individual is an instantiation of this abstraction which one numeric argument. By this argument, the instantiation reads the correspondent genotype text file. The initial arguments are used to pass its unique name to all six arrays belonging to each individual genotype. Using the ITD sound location technique, as described in section 2, we emulate the individuals dynamic position in a horizontal plan, as if they were moving inside a sonic location field. The casual encounter between individuals raises the chances of an offspring creation. This process entails to a varying-size population, different from the original ESSynth method, where the population had a fixed amount of individuals. Another distinction is that the output sound in ESSynth was given by the queue of each best individual from a population generation (audio samples of several ESSynth simulations are available at: www.nics.unicamp.br/~fornari). In this work, the sound output is given by all individuals coexisting at each moment with the variant population. This also makes possible the events of: population extinction; when its number of individuals decreases to zero, or super-population; where all computational resources of the machine running this Pd patch would be taken. We can set thresholds to avoid these two extreme scenarios but the natural variation of population size inside these limits is enticing and welcome for the perspective of this artwork.
The data from two groups of six tables corresponding to the genotypes of two active individuals are used by the genetic operators: crossover and mutation to create the group of six tables corresponding to the genotype of their offspring. These genetic operators are implemented in the sub-patch named "pd genetic operators", as seen also in Figure 4. This sub-patch receives two normalized scalar parameters, from the vertical sliders. As labeled, they are the coefficients of each genetic operator rate. These parameters go from zero to one, meaning that they vary from null to full genetic operation. They can be set on-the-fly by the user.

Similar to the ESSynth original method, this implementation also embodies the paradigm of variant similarity. However, with the sound output resulting from a dynamic population formed by all active individuals generated by the reproduction process, a soundscape will be naturally self-organized.

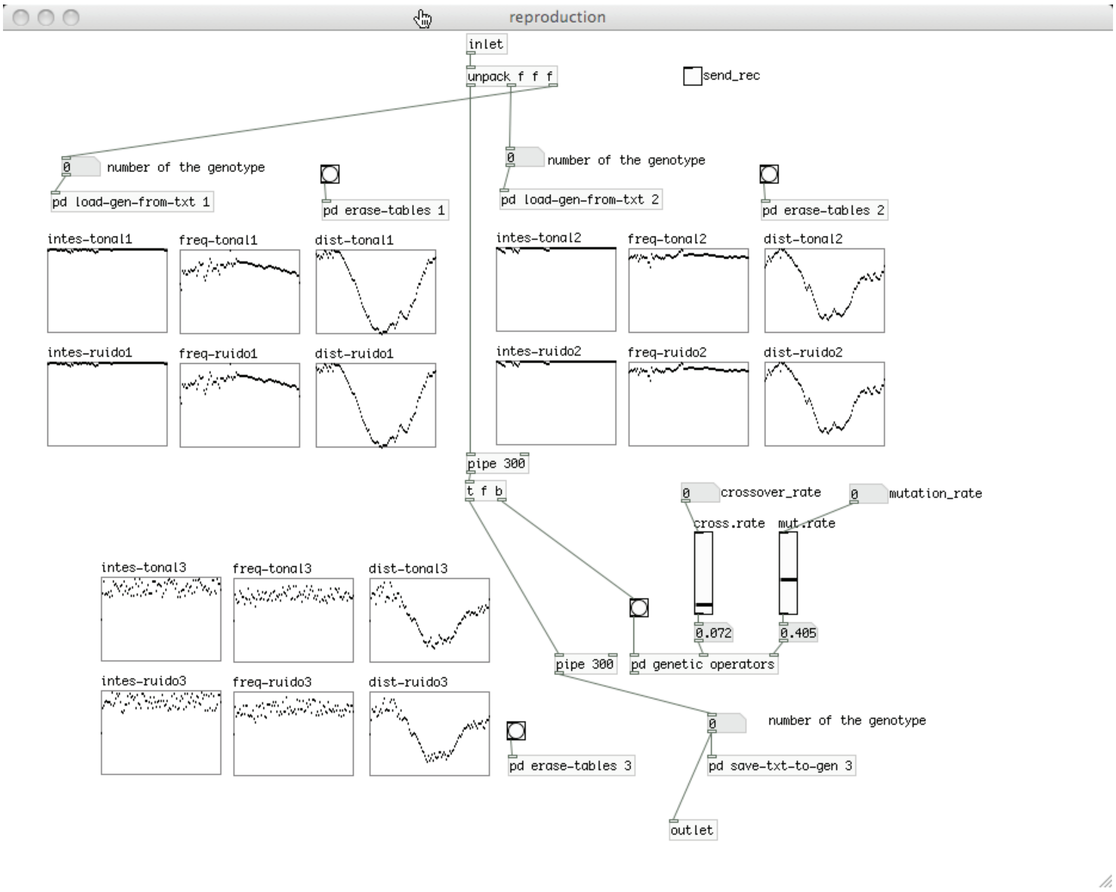


Figure 4. Implementation of the Reproduction process.

5. Conclusions

This work is about the implementation of an evolutionary soundscape system in PD that used as genotypes two types of artistic gestural data: one from the mappings of conceptual drawings and another from the time series retrieved out of dance movements.

The evolutionary system implementation made possible to indefinitely extend the duration of a processual artwork into the sound domain. We aimed to experience with these multimodal gestural data that seeded the evolutionary sound process in the creation of a soundscape resembling gestures that came from processes that originated art installations of other medias. The drawing collection, here seen as the registry in paper of gestures from a processual artwork, was finnished by the time we had collected its data, by the image mapping of of some drawings. Differently, the dance movements were collected as the dance performance was taking place. The evolutionary system took these gestural information and turned them into a soundscape that can exists for as long as the system is running, thus creating a dynamic population that does not repeat itself although retaining its sonic identity. Examples of the sound generated by this system, as well as the entire implementation in PD can be downloaded from the following link: www.nics.unicamp.br/~fornari/sbcm09.

The implementation of this system derives from the ESSynth method, in which there was a population of waveforms evolving in time guided by a Target set, representing the pressure of environmental conditions found among any biological population. However, in the implementation shown here, there is no Target set. This is due to the fact that, as the population is growing out of few individuals, we thought that a selection criteria could restrict the chances of this system to create a rich soundscape. Nevertheless, we plan to implement in further works a selection process based in some enhancements that will enable the external interaction with the dynamic soundscape creation process. We may experience with the usage of sensors and/or interface controls. Motion sensors and web-cams can easily be used with Pd patches to emulate external conditions that guide the evolutionary process. One topic that we plan to implement is the concept of energy intake, also known as “synthetic forager” where individuals can seek out and compete for food. The concept of individual gender is also still to be implemented. We plan to experiment with the notion of multiple gender individuals. We may as well implement a childhood period in the system, where individuals wont be able to go through the reproduction process but could receive information from other active individuals, somehow emulating a learning period. As seen, there is a myriad of interesting possibilities using the evolutionary method to create new implementations that can turn out into artwork installations, as well as being used by processual artworks exploring the multi-modality and interactivity, in order to reach immersive and adaptive sonic experiences.

6. Acknowledgements

We would like to thank the participation of Tatiana Benone, that recorded the body movements. We also want to acknowledge the financial support of Faepex/Unicamp, that sponsored this project. Manzolli is supported by the CNPq.

7. References

Lippard, Lucy. Six Years, Studio Vista, London, 1973.

Galenter, Philip. What Is Generative Art? Complexity Theory as a Context for Art Theory in: «Generative Art Proceedings», Milan, 2003.

Cramer, Florian. Words Made Flesh. Code Culture Imagination. Piet Zwart Institute, Rotterdam, 2005.

Truax, B. 2008. Soundscape composition as global music: Electroacoustic music as soundscape*. Org. Sound 13, 2 (Aug. 2008), 103-109.

Husarik, Stephen, American Music, Vol. 1, No. 2, pp. 1-21. Published by: University of Illinois Press. Summer of 1983.

Dexter, Emma (introduction). Vitamine D: New Perspectives in Drawing. Phaidon, London, 2005.

J. Blauert. *Spatial hearing: the psychophysics of human sound localization*. Cambridge: MIT Press, 1997.

V. Pulkki, "*Virtual sound source positioning using vector base amplitude panning*". Journal of the Audio Engineering Society, vol. 45, pp. 456-66, 1997.

J. M. Chowning, "The simulation of moving sound sources," presented at *Audio Engineering Society 39th Convention*, New York, NY, USA, 1970.

Jack B. Kelly and Dennis P. Phillips. “*Coding of interaural time differences of transients in auditory cortex of Rattus norvegicus: Implications for the evolution of mammalian sound localization*”. Hearing Research, Vol 55(1), pages 39-44. 1991.

Birchfield, S.T. & Gangishetty, R. “Acoustic Localization by Interaural Level Difference”. *IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), Philadelphia, Pennsylvania, March 2005.

Douglas S. Brungart and William M. Rabinowitz. "Auditory localization of nearby sources. Head-related transfer functions". The Journal of the Acoustical Society of America September - Volume 106, Issue 3, pp. 1465-1479. 1999.

Murray, J. C., Erwin, H. R., and S. Wermter, "Robotic sound source localization using interaural time difference and cross-correlation". *In proceedings of the KI-2004*, September 2004.

Holland J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology*, Control and Artificial Intelligence, MIT Press, Cambridge MA, 1992.

Holland J.H. *Hidden Order: How Adaptation Builds Complexity* , Addison-Wesley 1996.

Fels, S. S. and Manzolli, J. "Interactive Evolutionary Textured Sound Composition". 6th Eurographics Workshop on Multimedia, pp. 153-164. Sept. 2001.

Manzolli, J., Fornari, J., Maia Jr., A., Damiani F. The Evolutionary Sound Synthesis Method. Short-paper. ACM multimedia, ISBN:1-58113-394-4. USA. 2001.

Fornari, J., Manzolli, J., Maia Jr., A., Damiani F., "The Evolutionary Sound Synthesis Method". SCI conference. Orlando, USA. 2001.

Fornari, J.; Maia Jr. A.; Manzolli, J.. "A Síntese Evolutiva Guiada pela Espacialização Sonora". XVI Congresso da Associação Nacional de Pesquisa e Pós-graduação (ANPPOM). Brasília. 2006.

Fornari, J.; Maia Jr. A.; Manzolli, J.."Creating Soundscapes using Evolutionary Spatial Control". *In proceedings of the EvoMusarts*, Spring-Verlag,Valencia. 2007.

Caetano, M. Jônatas Manzolli, J. Fernando Von Zuben. F. Self-Organizing Bio-Inspired Sound Transformation. In *proceedings of the EvoMusarts*, Spring-Verlag,Valencia. 2007.

Von Foerster, H., "*On Self-Organizing Systems and Their Environments*." In: Self-Organizing Systems, M. C. Yovits und S. Cameron (Hg.), Pergamon Press, London, pp. 31–50, 1960.

Caetano, M. Jônatas Manzolli, J. Fernando Von Zuben. F. Self-Organizing Bio-Inspired Sound Transformation. In *proceedings of the EvoMusarts*, Spring-Verlag,Valencia. 2007.

R. Murray Schafer, M. "*The Soundscape*". ISBN 0-89281-455-1. 1977.

Truax, B. "*Handbook for Acoustic Ecology*". ISBN 0-88985-011-9. 1978.

Pforsich J., Handbook for Laban Movement Analysis. Written and Compiled by Janis Pforsich. 1977.

Mancini, M., Castellano, G.. Real-time analysis and synthesis of emotional gesture expressivity, in Proc. of the Doctoral Consortium of 2nd International Conference on Affective Computing and Intelligent Interaction, Lisbon, September 2007.

Maia, A., Lopes, J., Manzolli, J., Valle, R. ELEMENTARIDADES Projeto de Composição Algorítmica e Arte do Movimento realizado entre o NICS (Núcleo Interdisciplinar de Comunicação Sonora) e o GITD (Grupo Interdisciplinar de Teatro e Dança) da UNICAMP baseado no modelo físico do movimento das partículas elementares da matéria. 2001.

LaValle, S., Planning Algorithms. Cambridge University Press. 2006.

Zmölnig I., Pure Agents - Augmenting live patching. Piksel 08 Presentations. Talks & Presentations at StudioUSF December 4-7. 2008.

Real-Time Uses of Low Level Sound Descriptors as Event Detection Functions Using the Max/MSP Zsa.Descriptors Library

Mikhail Malt[1], Emmanuel Jourdan[1]

[1] Ircam – Institut de recherche et coordination acoustique/musique
1 place Igor Stravinsky, 75004, Paris France

{mikhail.malt,emmanuel.jourdan}@ircam.fr

Abstract. *This paper is a continuation of the research and development we began last year [Malt and Jourdan 2008] on the study and the use of audio descriptors for real-time performance of electroacoustic mix music, and as tools for computer-assisted musical analysis. Our main goal, in this paper, is to propose easy and efficient strategies for event detection in the context of real-time mix music (acoustic and electroacoustic music). We will examine three cases of the use of audio descriptors to build event detection functions: spectral slope, spectral standard variation and the construction of compound descriptors.*

1. Introduction: Event and Onset Detection in Real-Time Electroacoustic Music

Current compositional practice often involves the use of unconventional or extended instrumental techniques (e.g. multiphonics, blowing into an instrument, using key clicks as part of the instrument's vocabulary, etc.). In the context of the mix music itself, we find ourselves in a situation where our main task is no longer a matter of detecting simple pitch onsets, but dealing with considerably more complex event detections in difficult contexts. These new contexts may involve large variations in amplitude by register, considerable variation in terms of the onset of sound, the need to separate a sound from background "noise," and problems with multiple-microphone sound capture of instruments whose patterns of sound radiation are atypical (e.g. the bassoon). In addition, there are also situations where event detection based on a timbre variation (bisbigliando) is more complex and requires specific techniques. Taken as a whole, these factors complicate the task of event detection considerably.

In this paper we propose to speak more in terms of "event detection" than "onset detection" - we are concerned not only with note onset detection, but also any kind of musical events that could be perceived as a discontinuity within a musically static flow, such as sound inflections or noisy playing techniques.

The event detection methods currently available for real-time purposes are mainly based on amplitude, pitch or spectral variation. All of methods are strongly affected by background noise and strongly dependent upon such external factors as microphone type, distance from sound sources and especially room effects.

As onsets, sound inflections and event detection are dependent on a large number of signal parameters variations such as amplitude, spectral brightness, spectral standard variation, spectral slope, spectral onset slope, roll-off point, and spectral envelope. It is important at this point to mention that all these parameters are not fully independent and that some of them are highly correlated. Bearing this in mind, we will propose three experiments in real-time event detection based on the use of audio descriptors.

A large part of our assumptions and hypotheses derive from our own pragmatic observations in building event detectors for different musical contexts, and from our own systematic study of descriptor signals coming from different kinds of musical materials. Our intention in this paper has a more pragmatic and ongoing aspect to it, as well - we are looking forward to planning more systematic experiments to compare and to find the limits of the methods we are going to propose. In the meantime, very good reviews and comparisons of onset detection functions could be found in Bello [Bello and all 2005] and Collins [Collins 2005].

2. Event Detection by Spectral Slope

The spectral slope is an estimation of how quickly the spectrum of an audio signal decreases towards the high frequency range, and is commonly computed using a linear regression on the magnitude spectra. In the recent version of the *Zsa.descriptors* library, we implemented an algorithm which is slightly different from that proposed by Peeters [Peeters 2004, p. 14], but is analytically equivalent. It is based on covariance and variance computation of the frequency bins and squared amplitudes (energy) in an FFT frame.

$$slope[t] = \frac{\operatorname{cov}\left(f, a^2\right)}{\operatorname{var}(f)} \quad (1)$$

Where:

a is the linear amplitude vector frame,

f the frequency vector frame

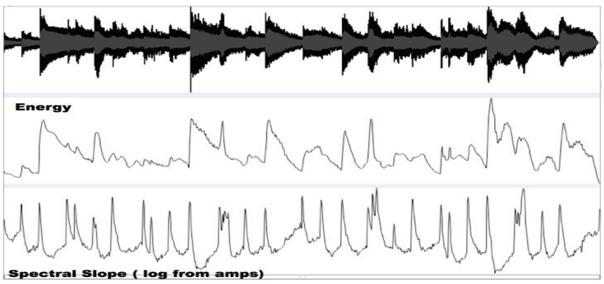


Figure 1. Onset detection function comparison (we have used the slope logarithm for the computation[1]).

What we have observed is that the spectral slope is less affected by amplitude variations and background noise than a signal coming from amplitude onset detection,

[1] This image was done with the *ftm.editor* from FTM library (© Ircam, IMTR team).

and has more well-defined peaks. Using the slope logarithm gives us even more precise peaks (Figure 1).

2.1. Preprocessing Technique

For certain applications we found it useful to introduce an adaptive signal level scaling in order to compensate for any level change (Figure 2). In fact, we used four parameters to control the preprocessing: a reference dB value (mean level to be maintained), a minimum dB to trigger the process (a signal below the trigger threshold will multiply the signal by zero) and a frequency to control a low pass filter to smooth the root mean square amplitude used to control the scaling and the gate. The final signal $S_{adapt}[n]$ will be:

$$S_{adapt}[n] = S[n] * K_{linear} \qquad (2)$$

Where:

$S[n]$ is the audio signal,

$A_{rms}(S[n])$ is the Root Mean Square amplitude of $S[n]$,

$A_{rmslp}[n] = aA_{rms}(S[n]) + bA_{rms}(S[n-1])$ is the smoothed $A_{rms}(S[n])$ signal, by a low pass filter,

A_{rms-dB} is the smoothed Root Mean Square amplitude of $S[n]$ in dB,

A_{ref-dB} is a reference dB value, this means, the average dB value we want reach with our input signal,

$A_{\min-dB}$, is the minimum level, in dB to trig the process,

$\Delta_{amp\_ref} = A_{ref-dB} - A_{\min-dB}$, is the is the dB difference between the reference value and the dB minimum reference value,

$\Delta_{amp} = A_{ref-dB} - A_{rms-dB}$ is the dB difference between the reference value and the actual signal amplitude, for

$$K = \begin{cases} 0, & \Delta_{amp} > \Delta_{amp\_ref} \\ \Delta_{amp}, & \Delta_{amp} \le \Delta_{amp\_ref} \end{cases} \qquad (3)$$

and $K_{linear} = 10^{\frac{K}{20}}$

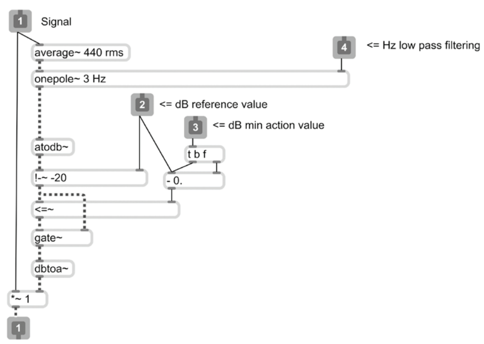
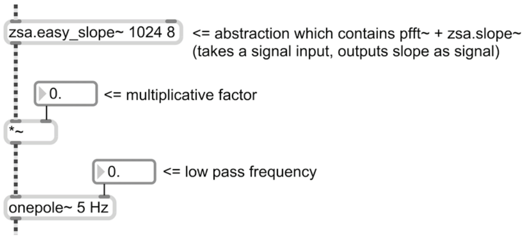


Figure 2. Adaptive signal level scaling, Max/MSP implementation

2.2. Building an Event Detection Function

The detection function was built using the *zsa.slope~* object [Malt and Jourdan 2008] (Figure 3) with a negative multiplicative factor (to turn the data positive) and a low pass filter to smooth out the data.



2

Figure 3. Spectral slope detection function

2.3. Peak Selection Technique

We propose a standard process shown in Figure 4 as a peak selection technique: A discrete derivative (we used samples steps for the derivative calculation) with a low pass filter to smooth the signal, a multiplicative stage to constrain the flow to {-1, +1} and threshold detection using standard Max/MSP objects (*thresh~* and *edge~*) with a final stage to control unwanted repetitions (see Figure 5).

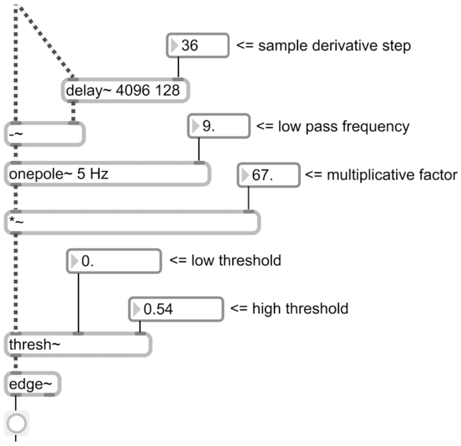


Figure 4. Peak picking technique

2.4. Avoiding Unwanted Repetitions

The avoidance of unwanted repetitions is critical, and can be easily implemented using the *onebang* object in conjunction with a *delay* object (Figure 5).

---

[2] In this paper we used encapsulated versions of "zsa.*" standard objects. The "zsa.easy_*" and "pfft~zsa.abs_*" abstractions are found in the Max/MSP zsa.descriptors library (http://www.e--j.com/?page_id=83).

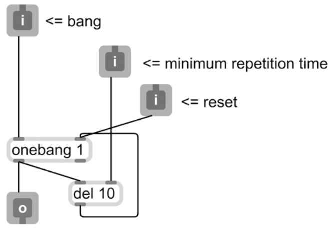


Figure 5. Delayed gate to avoid unwanted repetitions

2.5. Practical Use of Spectral Slope

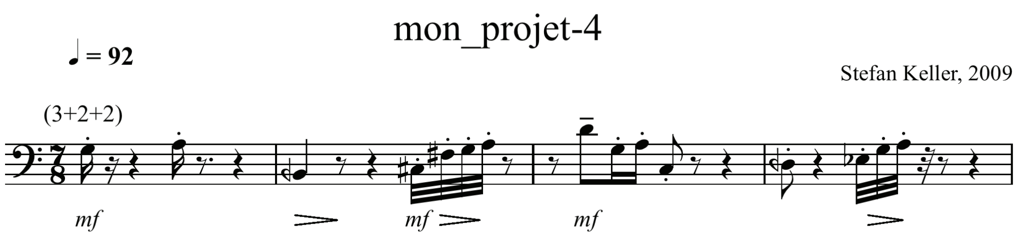


Figure 6. First measures of *mon_projet-4* from Stephan Keller

The sound radiation of the bassoon is heavily dependant on the register being played and therefore usually requires multiple microphones to obtain a satisfactory sound capture for the whole tessitura. Even with such a system, amplitude detection is further strongly affected by the movements of the performer. The spectral slope is correlated with the spectral envelope, and its variation is more related to the variation of the shape than it is related to the amplitude variation. Since we needed a highly compressed signal to obtain a good amplification of the instrument in this specific case, the calculation of the spectral slope clearly gave better results due to the fact that it was less dependant on variations in the amplitude of the signal from the microphone.

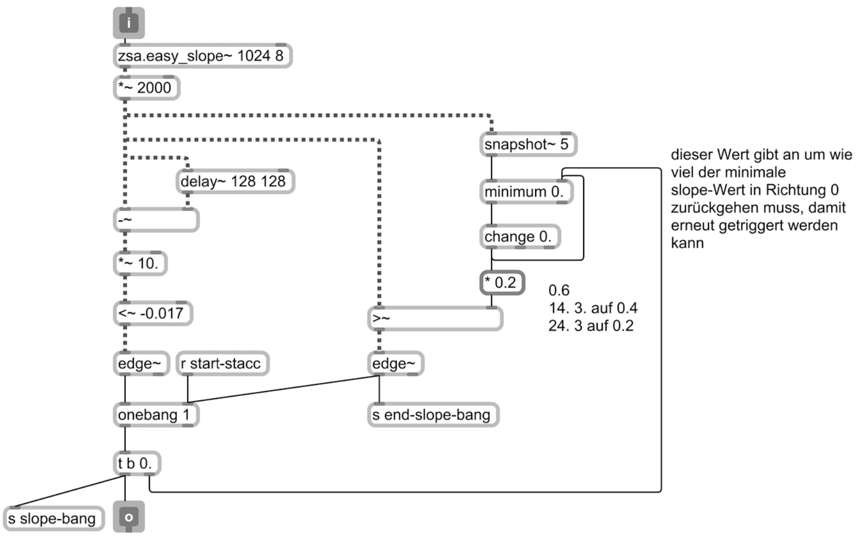


Figure 7. Spectral slope onset detection, implementation, by Stephan Keller

We found that the use of spectral slope also produced similar results when the Italian composer Franscesca Verunelli did some experimentation to detect accordion inflections going from very low dynamics to strongest ones as part of his work in the

Ircam Cursus. As was the case with the bassoon, the use of spectral slope proved to be more robust than a single amplitude follower and less susceptible to background noise.

3. Event Detection by Spectral Standard Deviation

Event detection functions that use spectral standard deviation have showed to be very useful as a technique for detecting noisy events in complex musical situations where different sound materials are mixed together. Any noisy event (e.g. key clicks) can be easily differentiated from more harmonic material; for example, a flute returns standard deviation values around 400-500 Hz (due to the player breath), while a key click returns standard deviation values around 1500-2500. An instrument that produces rich spectral data such as an oboe or violin will return standard deviation values in the1000-1500 Hz. Frequency range. A violin Bartok pizzicato could return values beyond 3000 Hz.

As you would expect, we use the spectral centroid as the first moment of spectra, considered as a frequency distribution, which is related with the weighted frequency mean value. The spectral spread is considered as the second moment - the variance of the mean calculated.

$$v = \frac{\sum_{i=0}^{n-1} \left(f[i]-\mu\right)^2 a^2[i]}{\sum_{i=0}^{n-1} a^2[i]} \qquad (4)$$

Where:

n is the half of the FFT window size,

i the bin index,

$a[i]$ is the amplitude of the bin i, in the real magnitude spectra of the FFT calculus and

$f[i]$ is the frequency of the bin i. Where:

$$f[i] = i * \frac{sample\ \ rate}{FFT\ \ window\ \ size} \qquad (5)$$

and μ is the spectral centroid in Hertz.

3.1. Building an Event Detection Function

We used the spectral standard deviation (the variance square root, gated by a "K" factor) to build our event detection function. The event detection function is defined as:

$$event\_Stdr\_function = \sqrt{v} * K \qquad (6)$$

Where the gate factor K is defined as follow:

$$K\left(A_{rms-dB}\right) = \begin{cases} 1 \ ; \ A_{rms-dB} \geq A_{\min-dB} \\ 0 \ ; \ A_{rms-dB} < A_{\min-dB} \end{cases} \qquad (7)$$

A_{rms-dB} is the rms smoothed signal level in dB and $A_{\min-dB}$ is the threshold dB value.

The use of the K gate is very important to avoid in silent musical passages, where the quick increase of standard deviation may be usually due to noisy flat spectra. The means of avoiding this problem is implemented in the *level_gate_scaling* subpatch (Figure 8, Figure 9)

Given the nature of the sound material and the presence or absence of interferences, it might be useful to smooth the standard deviation signal with a low pass filter (for situations such as detecting blow playing in woodwinds), but if we have a good sound capture and adequate percussive event detection, it could be avoided.

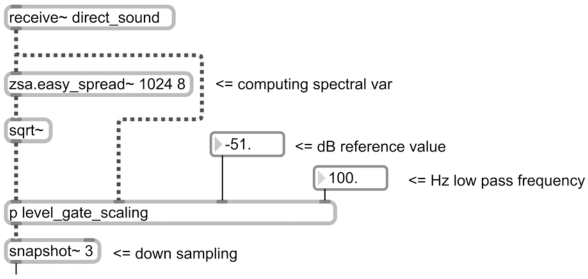


Figure 8. Gated spectral standard deviation, event detection function

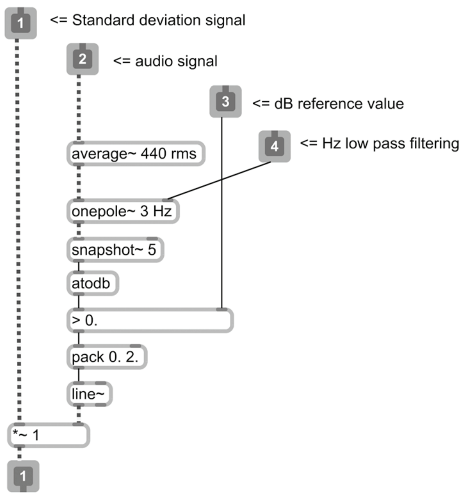


Figure 9. Level gate scaling

3.2. Peak Selection Techniques

For a peak-selection technique we used the same standard technique as for the onset event function, which uses the spectral slope (Figure 4, Figure 5).

3.3. The Practical Use of Spectral Standard Variation

The Italian composer Danielle Ghisi used this technique successfully within the context of the piece "*Comment pouvez vous lire à present ? Il fait nuit.*" (for Alto Sax and real-time electronics) created at Ircam's Espace de projection in March, 2009. The technique was used to detect key clicks in the last section of his piece in measures 90 to 94 (see Figure 10) using the spectral standard variation to trigger various real-time processes.

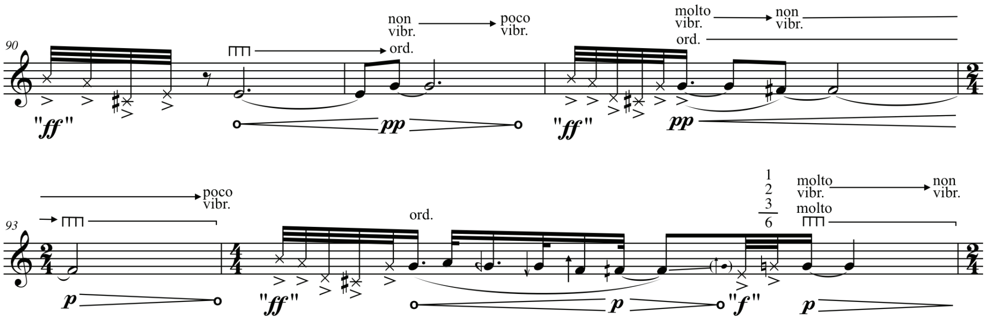


Figure 10. *Comment pouvez vous lire à present ? Il fait nuit,* measures 90 to 94, from Danielle Ghisi

Figure 11 shows the spectral standard deviation signal, the derivative and the onset detection of the first gesture (four key clicks followed by an E4-G4 in crescendo-decrescendo) of measure 90. The difference between the standard deviation from key clicks (around 1500-2000 Hz) and for the E4-G4 (around 150 Hz) is quite obvious and shows the advantage of this technique to detect noisy events.

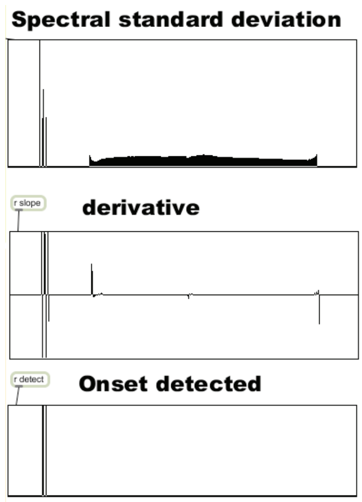


Figure 11. Visualization of spectral standard deviation event detection function, the discrete derivative and the event detection

4. Event Detection by Compound Function

The next experiment in event detection involves the use of a function based on a compound descriptor. In spoken sounds, we have observed that the fricative consonants tend to have a high centroid and a high spectral standard deviation.

$$D[n] = \left(\left(\frac{\mu}{c1} \right) * \left(\frac{\sigma}{c2} \right) * K \right)^2 \quad (8)$$

where:

μ is the spectral centroid,

$c1$ is a constant set in order to normalize the μ value,

σ is the spectral standard deviation,

$c2$ is a constant set in order to normalize the σ value and

K is defined as

$$K\left(A_{rms-dB}\right)=\begin{cases}1\ ;\ A_{rms-dB}\geq A_{\min-dB}\\0\ ;\ A_{rms-dB}<A_{\min-dB}\end{cases}\qquad(9)$$

The K variable is defined in order to avoid side effects in silent passages (see item 3.1). The main expression (7) maximizes the spectral centroid and standard variation.

4.1. Building an Event Detection Function

The expression (6) is implemented as it is shown in Figure 12, using a *pfft~* object as the core of the process, with two objects sharing the same FFT and energy calculation (Figure 13). To limit the noise impact, low pass filters smoothed the two descriptors signals.

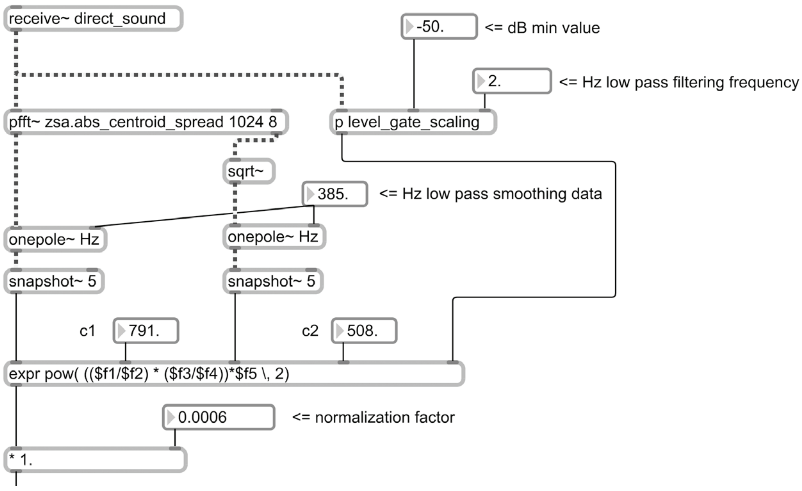


Figure 12. Compound detection function with centroid and spectral standard deviation

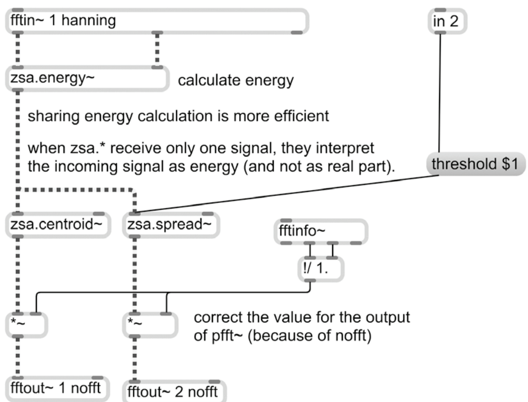


Figure 13. Patcher detail from *pfft~* object from Figure 12

Figure 14 shows the K variable implementation, where we used an RMS level smoothed by a low pass filter.

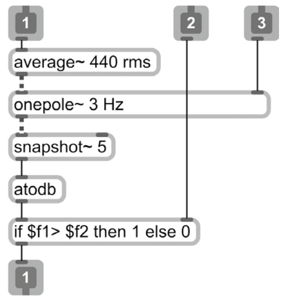


Figure 14: Patcher implementing the κ variable

4.2. Peak Selection Techniques

In this experiment, we used peak selection technique with an adaptive threshold (Figure 15).

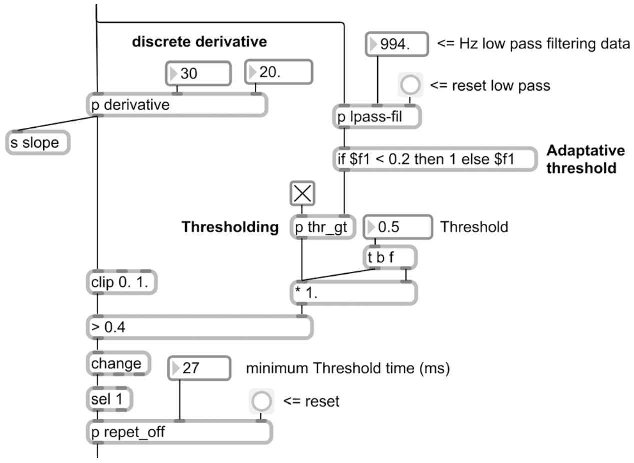


Figure 15. Adaptive threshold

First, the data flow coming from the detection function is derived (in a discrete way, see Figure 16), returning the function $D'[n]$.

$$D'[n] = \frac{\Delta D[n]}{\Delta n} = \frac{D[n+m] - D[n]}{m} \qquad (10)$$

Where:

$D[n]$ is the event function at index n, and m is a window size.

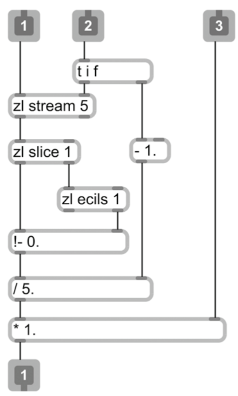


Figure 16. Discrete derivative from expression (10)

The result passes by a threshold process where the reference threshold value δ_{ref} is weighted by a smoothed version of the detection function. We used a low pass filter (Figure 17). The adaptive threshold took the shape of expression (9)

$$\delta[n] = \delta_{ref} * \left(aD[n] + bD[n-1]\right) \quad (11)$$

Where:

$\delta[n]$ is the adaptive threshold at time n,

δ_{ref} is the reference threshold value,

$D[n]$ is the descriptor value from expression (7) at time n, and

a,b are the coefficients for the low pass filter.

Our final detection function $\sigma[n]$ will take values of zero or one.

$$\sigma[n] = \begin{cases} \sigma[n] = 1; D'[n] \geq \delta[n] \\ \sigma[n] = 0; D'[n] < \delta[n] \end{cases} \quad (12)$$

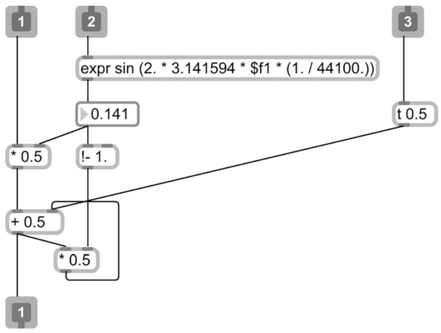


Figure 17. Low pass filter from expression (11)

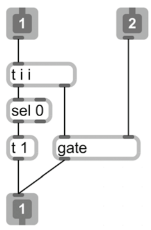


Figure 18. Gate to choose between adaptive or static threshold

5. Conclusions and Perspectives

The use of sound descriptors appears to be a useful and advantageous tool for musical event detection in real-time music. The methods we described were used as pragmatic alternatives to event detection in complex cases where the usual techniques for event detection did not perform sufficiently well. In those cases, the use of low-level spectral audio descriptors proved to be satisfactory and robust. These promising results strengthen our motivation to continue our research in this direction.

We are looking forward to improve the number of *Zsa.descriptors* modules, to add high-level descriptors, and develop to systematic means to compare the efficiency of the various event detection functions for real-time process.

6. Acknowledgments

We would like to thank Gregory Taylor and David Coll for their valuable remarks, comments and suggestions.

7. References

G. Peeters, A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Cuidado projet report, Institut de Recherche et de Coordination Acoustique Musique (Ircam), 2004.

M. Malt and E. Jourdan, "Zsa.Descriptors: a library for real-time descriptors analysis", in 5th Sound and Music Computing Conference, Berlin, Allemagne, 31th july to August 3rd, 2008, p. 134-137. See http://www.e--j.com/?page_id=83.

X. Rodet and F. Jaillet, "Detection and modeling of fast attack transients," in Proc. Int. Computer Music Conf., Havana, Cuba, 2001, pp. 30–33.

I. Kauppinen, "Methods for detecting impulsive noise in speech and audio signals," in Proc. 14th Int. Conf. Digit. Signal Process. (DSP2002), vol. 2, Santorini, Greece, Jul. 2002, pp. 967–970.

D. Schwarz, Data-Driven Concatenative Sound Synthesis, PhD Thesis in Acoustics, Computer Science, Signal Processing Applied to Music, Université Paris 6 - Pierre et Marie Curie, January 20, 2004.

J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M.B. Sandler, "A Tutorial on Onset Detection in Music Signals", in IEEE Transactions on Speech and Audio Processing, Volume 13, Issue 5, Sept. 2005, p. 1035 – 1047.

N. Collins, "A Comparison of Sound Onset Detection Algorithms with Emphasis on Psychoacoustically Motivated Detection Functions". Proceedings of AES118 Convention, 2005.

A. Klapuri, "Sound Onset Detection by Applying Psychoacoustic Knowledge", in Proc. IEEE Conf. Acoustics, Speech and Signal Proceeing (ICASSP,'99), 1999.

X. Rodet and F. Jaillet, "Detection and modeling of fast attack transients", in ICMC'01, La Habana, Cuba, Sept. 2001.

L. O. Nunes, P. A. A. Esquef, and L. W. P. Biscainho, "Evaluation of Threshold-Based Algorithms for Detection of Spectral Peaks in Audio," in Proc. 5th AES-Brazil Conference, São Paulo, Brazil, May 2007, pp. 66-73.

Music Creation by Novices should be both Prototypical and Cooperative - Lessons Learned from CODES

Evandro Miletto[1]*, Marcelo Pimenta [1], François Bouchet[2], Jean-Paul Sansonnet[2], Damian Keller[3]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul
Caixa Postal 15064 – 90501-970 Porto Alegre, RS. Brasil.

[2]LIMSI-CNRS.
BP 133, 91403 Orsay cedex, France.

[3]Núcleo Amazônico de Pesquisa Musical - Universidade Federal do Acre
Caixa Postal 500 – 69915-900, Rio Branco, AC. Brasil

{miletto,mpimenta}@inf.ufrgs.br, {bouchet,jps}@limsi.fr, dkeller@ccrma.stanford.edu

Abstract. *Musical creation is usually considered as mostly a solitary activity done by composers but we are convinced that CODES - a Web-based environment designed to support Cooperative Music Prototyping (CMP) - offers great contributions to social ways of music creation by novices. One of the main findings obtained during CODES development and usage is that systems aiming at providing effective support to such music creation activities by novices should meet specific requirements, in order to support a dynamic and creative environment, enabling knowledge sharing by means of rich interaction and cooperative mechanisms adapted to address the idiosyncrasies of this CMP context. The goal of this paper is to present, discuss and illustrate these prototypical and cooperative aspects of novice-oriented music creation activities in CODES.*

1. Introduction

> *"The first question I ask myself when something doesn't seem to be beautiful is why do I think it's not beautiful". – John Cage.*

Music has been described as a social activity in which we share a musical experience [Gurevich, 2006]. Clearly, technology has created new social modalities for music listening, but we are convinced that technology also offers great contributions to social ways of music creation.

Musical composition is a complex activity where there is no agreement about which activities have to be performed and in which sequence: each person (composer or not) has a unique style and way of working. As a consequence, most of composers have not developed yet the tradition of sharing their musical ideas and collaborating while composing and thus composing is considered as mostly a solitary activity done by composers. Music creation is a *design activity*: the design of (new) sounds and/or the design of (new) combinations of (existing) sounds, forming (new) sound sequences or simple musical pieces. However, novices in music (here called *novices*) do not have enough knowledge and confidence to create music by themselves: usually they do not have access to musical

*Supported by Instituto de Informática - UFRGS, CNPq and CAPES.

instrument and do not know how to play them, neither how to represent music using traditional notations. Indeed, novices in music need effective interactive support to cooperate with each other for producing music, instead of only consuming it.

Even presenting some constraints as for sound information traffic, the Web is becoming increasingly attractive as a platform to support both social activities and music making [Iazzetta and Kon, 1998]. For example, today YouTube [Google, 2009] and other social Web services, such as MySpace [Media, 2009] and Flickr [Yahoo, 2009], have improved the interaction between users and systems over the Web, and users are getting used to new purposes, like engagement and self-expression. In fact, Web 2.0 has turned the passive user into an active producer of content and shaper of the ultimate user experience, and the Web is becoming a rich and ideal environment for social activities.

CODES is a Web-based environment designed to support Cooperative Music Prototyping (CMP), with special focus on novices in music. But, differently from YouTube, Flickr, and even MySpace, where people only publish their content, Web systems for experimenting with music should also provide ways to create contributions and experiments. For this reason, we consider CODES as a system for *music creation*, instead of a system just for publishing music. CODES offers a high level music representation and user interface features to allow easy direct manipulation (drag-and-drop) of icons representing sound patterns.

The main motivation of our work is the belief that no previous musical knowledge should be required to any ordinary user for participating in any experiment of music creation. Of course, it is not a matter of musical quality of the finished work, but the mere possibility of "creating it". Clearly, novices are not composers but we assume that they may be able to do musical creative work if they have an adequate support.

In this paper we present two very important principles, learned and confirmed by findings obtained during CODES development and usage, to be considered when providing such support to novice-oriented music creation activities: a) Music creation by novices should be prototypical; and b) Music creation by novices should be cooperative.

A prototypical music creation process means novices can draft simple musical pieces - we call them *Musical Prototypes (MPs)* in order to highlight the difference - which can be tested, modified, and repeatedly listened to, in a cyclical refinement of initial musical sketch until a final stage being reached. This process clearly resembles prototyping cycles adopted in industry and in incremental software development. Since music creation is in fact a (music) design activity, it seems natural and straightforward to adopt a prototypical process. In the music literature, "draft" is commonly applied to such kind of creative work, but here the emphasis is focused on the cyclical prototyping process and not on the product itself, and consequently in this paper "prototype" and "draft" correspond to the same idea.

In a cooperative music creation process, the refinement of an initial musical idea is consequence of a collaboration of the author(s) of initial musical idea and of their partners, all members of a group (in fact, a social network built by explicit invitation) that will be cooperating until a final consensual stage of MP be reached. This process is noticeably a particular kind of Human Centered Collaborative Design where the result of design is a MP.

Through the prototypical and cooperative nature of CODES, novices may thus have the opportunity to be, like experienced musicians are, the actors of their own musical experiences. It implies new requirements that should be taken into account when we consider these novices as a new user profile: the Web composers [Miletto et al., 2009], e.g., someone actively participating in a CMP.

The goal of this paper is to discuss the prototypical and cooperative aspects of music creation activities in CODES, focusing on presenting and illustrating its features designed specially for novices in music.

The paper is structured as follows. The next section resumes how CODES made music creation possible by novices in music. In Sections 3 and 4, we discuss the reasons why novice-oriented music environments should be prototypical and cooperative, respectively. Section 5 presents the particular viewpoint of the cooperative activities in CODES. An evaluation showing what actual novice users are saying about CODES, along with the results of our experimental work is presented in Section 6. Finally, in Section 7 are the conclusion and final remarks of this paper.

2. Music Creation by novices: how CODES made it possible?

CODES - COoperative Music Prototype DESign, is a Web-based environment for cooperative musical prototyping that aims at allowing novice users to experiment with music and interact with each other in order to create musical prototypes.

To make it possible, CODES enables novices users to perform four main tasks in a high level of interaction. Such tasks include *creating*, *editing*, *sharing*, and *publishing* musical prototypes. Users can create a new MP by clicking in this option (see Figure 1. a), choosing its name, and optionally its musical style. Since all the styles are available at the sound library, mixing sound patterns from different styles in the same MP is possible.

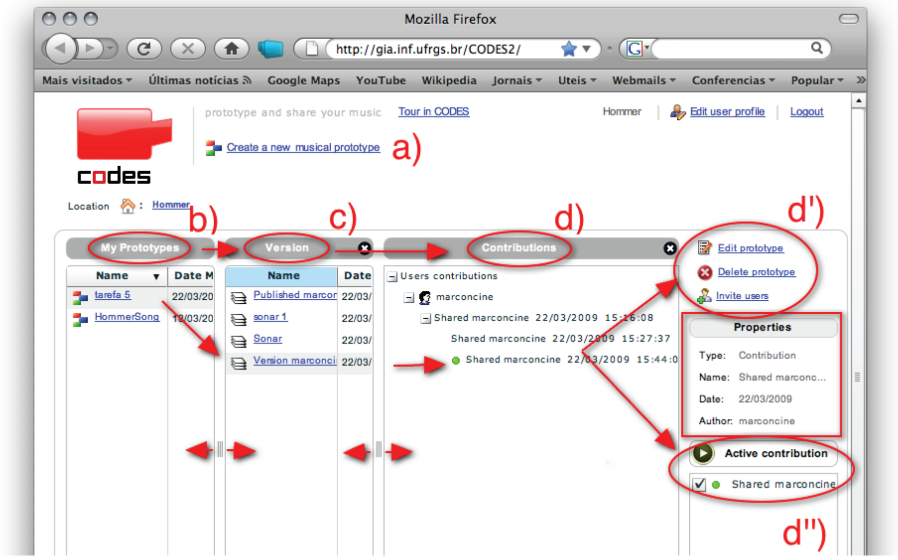


Figure 1: Excerpt of the screen which list the users' musical prototypes

Figure 1 shows how CODES organizes the users prototypes list. Users can see their MP information in a kind of hierarchical structure by clicking in one of the list (Figure 1.b). Each MP can have one or several versions (Figure 1.c), which also can have one or several contributions (Figure 1.d). Such contributions can be selected and combined to be listened to (Figure 1.d") or edited (Figure 1.d').

Edition in CODES includes actions of manipulation of sound patterns from the sound library to the editing area, such as "drag-and-drop", "delete", and "expand" the duration to listen to the final result. See a screenshot of the CODES editing level in Figure 4.

For sharing a musical prototype, the user "owner" can invite CODES users using a search engine or sending explicit invitations by e-mail to non-members and asking them for cooperation as illustrates the Figure 2. When someone accepts such an invitation, this user becomes a prototype partner and can edit it like the owner does.

At any time users can listen to the musical prototype and link arguments to their decisions, in a similar structure of a design rationale. Thus, all prototypes' partners can discuss and share ideas about each step of the prototype refinement, in order to understand someone else's decisions.

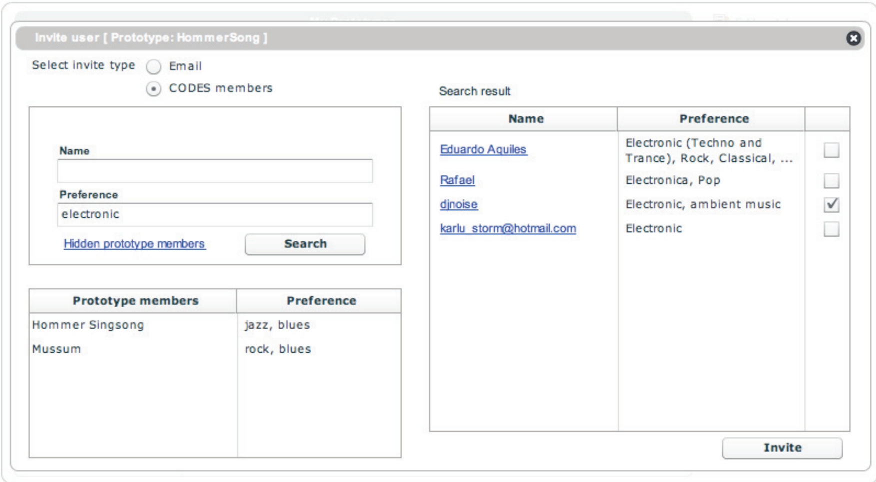


Figure 2: Screenshot of CODES inviting window

When someone considers the result sounds good, a "publication request" can be triggered and the group may discuss and deliberate about the publication of this musical prototype in the CODES home page. This activity is named *musical prototype publishing*. As an alternative to publishing their music, users may export (download) their musical prototype in an MP3 file format and share it as they want.

Throughout the design of CODES, we sought to emphasize two principles which summarize the lessons learned so far, emerged during CODES development and confirmed by evaluations of CODES users in actual usage, to be considered when providing support to novice-oriented music creation activities : a) It should be prototypical; and b) It should be cooperative. We chose to explore these principles because they have received little explicit attention within the networked music domain, mainly in a novice-oriented perspective. We will discuss these principles more deeply in the following sections, also presenting and illustrating how CODES takes them in account.

3. Novice-oriented music environments should be prototypical

Like Weinberg [Weinberg, 2002], we are interested in providing specially to novices an access to meaningful and engaging musical experiences. In CODES musical prototyping process MPs are repeatedly tested, listened to, and modified by their first author and their online partners, until their final forms are reached. Like any other design-related prototyping process, CODES music prototyping process is iterative, incremental and evolutionary, since an initial musical idea (first version of a MP) is produced and refined through a number of stages up to the final version. Moreover, this refinement help users to discover, validate, or derive new musical ideas from their initial musical ideas. We believe this prototyping process is one of the most interesting aspects by using CODES. It enables actual creation and experimentation (hearing) of musical ideas, by means of the rich interaction mechanisms - designed to improve user engagement with the system - associated to each prototype edition and modification.

In order to create a new prototype using CODES, user needs only to select pre-existing sound patterns from a sound library and group them in an arbitrary way (i.e where and how the user wants). Before selecting a sound pattern, the user can play it to check if it is the right choice. The user can edit (inserting, removing, resizing, changing order of sound patterns) and also play the MP at any time as desired (the sounds displayed in the editing area are played from left to right), in a dynamic and interactive way. The freedom for the MP manipulation and experimentation is a basis for a successful music prototyping process. Indeed, as our intention was to design a musical environment where such music prototyping process seems natural to users and the support provided to such process could be considered useful and usable mainly for novices, CODES interaction design was approached from an HCI perspective.

In an HCI perspective, any design should start by a study aiming at identifying and knowing the users, their goals and what tasks they need to perform to achieve them. Since the target user group is composed by novices in music, it is very difficult to define clearly goals and tasks based on existing software applications for music creation. Indeed, most of them and their user interfaces features are only suitable for musicians, not for novices.

First of all, musicians know music theory. They know how to read scores, the traditional music notation with its staff, and musical symbols. Moreover, they know these symbols refer to concepts like notes, rests, and tonalities - a novice may not even know what these musical concepts are all about! Even alternative notations (like tablature) contain alternative symbols for the same concepts, and the problem remains: these concepts are not part of a novices' world. Notation is a hard and non-intuitive concept for any novice to learn. In addition, musicians also have theoretical and practical knowledge about musical instruments, have access to them, and know the technical issues related to how to play them.

As a consequence, usual music software often relies on traditional music representations and on metaphors from a musician's experience. The MIDI protocol itself, which is designed to interconnect digital musical instruments and computers, is based upon "musical performance event", like keys being pressed, changes in timbre and in tonality, tempo changes, etc. Even some more recent interaction styles (like for example the style adopted by IRCAM's Max/MSP [Cycling74, 2009]) are metaphors of something musicians are used to do, requiring experienced musician's knowledge and vocabulary, and they are consequently inadequate for novices.

Since we did not have similar environments to use as a basis, we have adopted an incremental and iterative design approach: to identify novices' needs, to design and fit the system to the users and their needs, to evaluate the design and use the evaluation result as a feedback to iterate until a good design can be achieved. The prototyping process resulting is cyclic and the interaction provided to novices is simple but rich. For example, to edit a MP in CODES is a very simple task: sound patterns are dragged from the sound library - always visible, and dropped into the MP editing area.

The CODES user interface has three main levels of interaction for different user profiles: a) Public Level, b) Musical Prototype Editing Level, and c) Sound Pattern Editing Level (see Figure 3 for an excerpt of the screens representing each of the levels).

Basically, the two different user roles are CODES *members* (registered users) and *non-members* (general public, non-registered visitors). The non-members are typically the Web users which can access the CODES home page shown in Figure 3.a) and listen

Figure 3: Screeshots of the three main levels of CODES

to the published musical prototypes, rate them, and search musics by author or style.

Once logged in CODES, members can interact with the two other levels shown in Figure 3.b) and 3.c) and find/invite partners in order to cooperate and share their musical ideas, edit musical prototypes, and related conversation/argumentation as describes the next section.

4. Novice-oriented music environments should be Cooperative

Since Web is nowadays a very common platform for social and collaborative activities, CODES project has moved the attention focus from individual to cooperative music prototyping process. Indeed, novices in music may not have enough knowledge and confidence to create music by themselves: by means of interactions with and advices from more experienced users, novices can improve their learning during the development of a collective music prototype. CODES provides an effective interactive support to make novices cooperate with each other for creating music.

Thus, a music prototype may be the result of an individual process of musical prototyping but also the result of a collaborative design, having the participation and contribution of many users. In this case, the MP is an artifact shared by all members and they have to cooperate by means of actions to manipulate the shared artifact (MP) and by means of explicit conversation and argumentation. The process of group formation and participation in group activities in CODES is simple: invitations can be sent as shown in Figure 2. Once logged in the system, users can send explicit invitations to other users (registered or not). For sharing a musical prototype, the user "owner" can invite CODES users using a search engine or sending explicit invitations by e-mail to non-members and asking them for cooperation. The *members list* and *group status* are also displayed in the members area. When someone accepts such an invitation, this user becomes a prototype partner and can edit it like the owner does.

At any time users can listen to the musical prototype and link arguments to their decisions, in a structure of a design rationale [Shum, 1996]. Thus, all prototype partners can discuss and exchange ideas about each step of the prototype refinement, in order to understand someone else's decisions. Each author's contribution in the shared workspace is identified by color: for example, the edges of sound pattern icons are colorful (the color obviously chosen by the user, as illustrates Figure 4.a'). In the members area (Figure 4.a), a user may show or hide other users' contributions (in fact, the other users' layers as shows 4.b) by clicking over the user id. It is possible to listen to each layer separately, to compare and combine contributions, and, of course, to save the result. The group status shows when there are new comments or new versions with icons. Figure 4 shows an example of three users cooperating in the same MP.

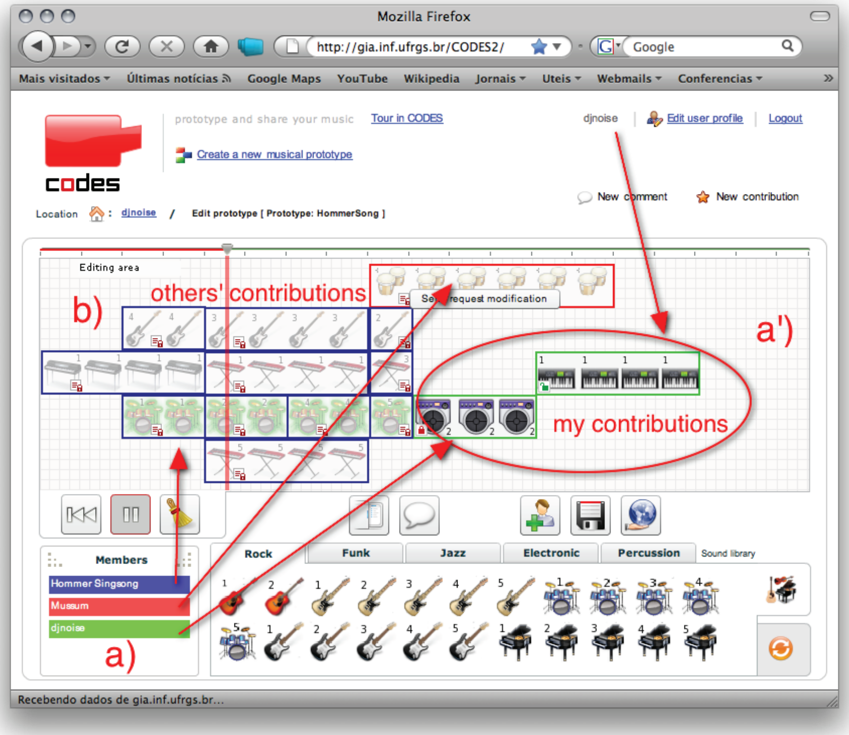


Figure 4: Three users cooperating in a shared MP at CODES MP Editing Level

When someone contributes by adding a new sound pattern to an MP, it will be, by default, locked for other users, with a blurred appearance. If some user wants to prototype or edit the other's locked layer (or sound patterns), CODES offers a special mechanism called "modification request".

The Music Prototyping Rationale (MPR) mechanism is another effective way to represent and to record explanations and argumentations for each action or decision made during CMP. Each user may associate comments and arguments (in favor or against) to any action on any prototype element. Each argument is related to a user or the whole group and the current layer.

In CODES, the basic elements of the MPR are "issues", "positions" and "comments". Issues correspond to decisions or actions that have been made or states which have been reached during an MP creation and refinement. A Position is a statement or assert that resolves the issue. In the case of CODES, positions can be pros, cons, idea, and important. Comments are asserted in order to agree with a specific course of action (comments in favor) or to express some objection (comments against).

Besides, CODES also adopts the notion of awareness, which is the understanding of the actions of other users providing to each user a context for his own actions. CODES offers three kinds of awareness mechanisms:

- MPR, to allow users to know the reasons behind other members' actions;
- Modification Marks, to indicate to a user that a prototype has been modified by others; and
- Version Control with layers, to keep an explicitly recorded track of the steps that led to the current prototype state.

CODES uses modification marks as the awareness mechanism to alert new events to a user, like modifications on a prototype or suggestions made by others.

4.1. The particular viewpoint of cooperative activities in CODES

Music creation by novices needs to provide a very specific kind of support for collaborative activities. In fact, the conventional cooperative approaches with fixed goals and roles, not allowing unsystematic and opportunistic negotiation are not adequate for the dynamic, creative, and collaborative nature typically related to collaboration in the arts, like CODES's Cooperative Music Prototyping (CMP).

Table 1 summarizes the main differences we have found comparing the cooperative support we have defined in CODES development with some characteristics generally found in cooperative environments for technical product design, like in manufacturing, construction, etc.

Table 1: Cooperative activities general framework

	Collaboration for Technical Product Design	**Collaboration in Non-Technical Product Design**
Main Goal	Product design according to a product (or requirements) model	Creative product resulting of a mutual understanding; no previous product modeling; mutual learning process is as important as the product itself
Group Topology	Typically, a hierarchical group with a leader	Typically, a non-hierarchical group without formal leaders
Control	Coordination	Argumentation
Planning and Decisions	Rigid and systematic plan definition, decision order following planning	Unsystematic and opportunistic negotiation
Roles and Tasks	Fixed roles with responsibility assignment; pre-defined task allocation	No fixed roles, no responsibility assignment; flexible task allocation
Example	Collaboration in manufacturing, construction, etc	Collaboration in the arts, cooperative music prototyping (CMP)

For technical products, there is a need for specifying a product model in order to standardize the process and predict the final result. For non-technical products, like music, the emphasis is on the subjective aspects of the act of creation rather than on following a model for creation. As we do not know a priori about the final result, the process is guided by the creation or creativity itself, instead of a previously defined design. As this process emerges from the cyclic interactions of the group, based on contributions from/to each other, the "control" of the process is done by negotiation between members, without the need for the role of an explicit controller. Thus, the "decisions" are supposed to be consensual by negotiation, and not imposed by the authority of a leader. We believe that it is not necessary to make a distinct and explicit representation of the leader, because usually in a hierarchical group, the leader's opinions and actions may inhibit the other users' participation. Indeed, interactions can evolve as time passes, and the more "skilled" users can be recognized and respected naturally by the group while suggesting and justifying their contributions. This allows total flexibility without needing prior role definition, task allocation or responsibility assignment for members.

Because the cooperative music prototyping process can justifiably be seen as a political process determined by conflicts and cooperation, the joint development of ideas by means of both multi-perspective approach and negotiation support is particularly important. The multiple actors - all who are cooperating in the refinement of the musical prototype - hold different perspectives on the creative process and its results (the musical prototype), each one with different backgrounds and opinions due to the context they come from. Therefore, it is essential to support mutual understanding and to resolve conflicts during cooperative musical prototyping.

A negotiation between these different viewpoints and goals must be explicitly supported and maintained over time, thus the decision-making process is cooperative and distributed. Real cooperative activities are very difficult to automate and to control because they involve the complexities and the dynamic nature of human group work, but we may attempt to support them.

In fact, we think this support for cooperative music prototyping is a particular kind of Human Centered Collaborative Design. The basic idea of our CMP process is that members cooperate not only by means of explicit conversation and explicit actions on a shared objects space, but also by interpreting the messages and actions of other actors in accordance with the model of their thinking and acting, which has been built up in the course of their interaction.

A shared objects space involves prototype-oriented information, which comprises all information about musical prototypes, including their composition (combination of sound patterns, versions formed by layers) and social-oriented information (including interactions between actors during the process).

Sound patterns are predefined 4s MP3 samples in the CODES sound library available for users represented by an iconic format. Manipulation of prototype oriented information is goal-motivated with typical prototype element manipulation, including use, modification, combination, replacement, and experimentation (audio listening) of sound patterns. Social-oriented objects are all related to conversation, like messages and comments. One significant consequence of recognizing social-oriented objects as relevant information is that, instead of considering modifications as only explicit transformations on an MP, we also consider the changes on social-oriented objects. That is, we interpret modifications on shared objects space as meaningful changes in both MP and social context.

This way, a sequence of messages may, at the same time, not change the MP and significantly alter an actor's argument, opinion or decision. Unlike a conversation, where messages are categorized by their purpose within the conversation, for action, for clarification, for orientation, and so forth, conversation in CODES is simply composed by all recorded messages sent and received to/from the CMP actors, indexed to other relevant model components. Then, recording the actors' messages is extremely useful to capture, in an implicit way, the background knowledge, concepts, definitions, and opinions surrounding their viewpoints.

In addition, the music prototyping rationale may be explicitly recorded by decisions and arguments. Decisions are goal-motivated consensual choices, concerning alternatives of the action course. Arguments are consensual explanation, not an individual message interchanged between actors. Every decision or action may be linked to (pro or con) arguments.

Notice that the actors cooperate via the shared objects space, that is, either indirectly by means of musical prototypes they manipulate and modify, or directly by means of conversation. Thus, the set of actions an actor may perform has been broadened to include direct interactions with other actors, in addition to traditional actions of prototype manipulation and the communication between group members plays a crucial role to support cooperative activities.

5. What are actual novice users saying about CODES: Evaluating Novice-oriented CODES features

CODES have been made available for use by actual users in our academic context. Following well-known subjective evaluation methods from the HCI field, we made some experiments to obtain qualitative results from the use of the CODES environment and its functionalities. Our goal was not only to get overall feedback (mainly subjective) from users but to try out our proposals for non-technical cooperative design environments as well.

In a first experiment, five individuals, representative of the CODES typical users (with ages from 20 to 35 years, having no musical expertise, and using CODES for the first time) had to perform fifteen real tasks. See Figure 5 for an illustration of the opinion poll characterization.

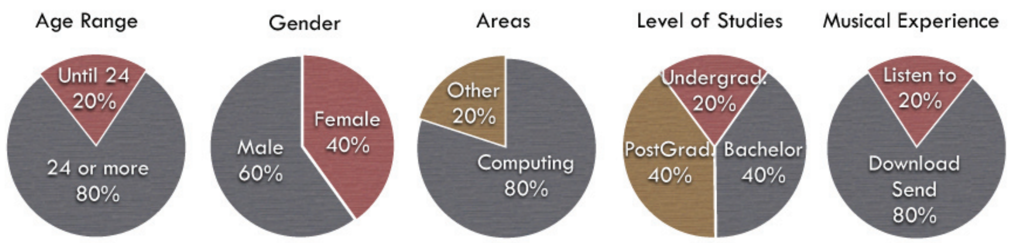


Figure 5: Opinion poll characterization of the experiment

These fifteen tasks were designed to simulate a scenario in which a novice user would learn how to create, edit and cooperate in a musical prototype. Particularly, a cooperative scenario was composed specifically by three different tasks at the MP editing level. The tasks included creation, edition, and sharing of CMP. Time taken to complete all the tasks ranged from 20 to 50 minutes.

The experiment carried out was the User Testing [Rubin, 1994] and was conducted in the presence of a facilitator (observer), a usability expert. He just read each task for the user, and took notes of any problems found and any verbal comments from them. The subjects were instructed to talk what they thought while interacting with CODES, thus using the "thinking aloud" method [Nielsen, 1992]. Interaction and user comments were also recorded with a video camera aimed at the computer screen (see Figure 6).

Figure 6: A recorded session of user testing with CODES

After performing the tasks, users filled out a form with open and closed questions.

The open questionnaire posed seven questions concerning the Nielsen's heuristics[1] like visibility, contextualization, control and freedom, feedback, flexibility, and the musical representation in CODES as well.

To answer to the eleven closed questions, a main question was made: *Do you agree with the following sentences?* Thus, the subjects should choose one of the five options: *Totally Agree, Agree, Neutral, Disagree, Totally Disagree.*

Different subjects were tested in order to get overall feedback of the users regarding usability, accessibility, and cooperative issues. Besides, the tests aimed at discovering interface and interaction drawbacks. Some users have also detected important drawbacks concerning the system feedback, according to the following quotes:

> "Sometimes, the system would give more feedback". "I do not know what is the session I am posting the comment". "I did not know why should I choose a color when registering myself in the system". "What means this icon in the editing area?"

Despite these negative points, overall test results were favorable and most of them have assigned "totally agree" as shows the Figure 7.

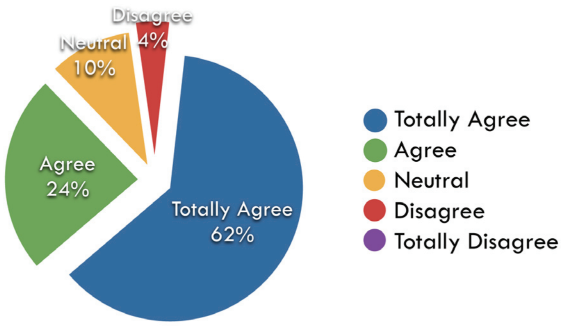


Figure 7: CODES General approval

The most important test results were those that allowed us to identify some user needs that where not being addressed by the initial design of CODES, and so would imply new requirements for the system. For instance, users would like to do their contributions and to combine them with any of the others, but without changing each participant's original and previous contributions. This was not possible in an early CODES version. This test result is the origin of a layer-based approach, described in [Hoppe et al., 2009].

The experiments were intended to be developed in a very restricted context, but until now it is possible to conclude that the system is intuitive and easy to use making users feel motivated by using CODES for enhancing and sharing their musical experiences.

6. Conclusion

In this paper, we have presented some principles emerged from (and adapted for) CODES to address the idiosyncrasies of this novice-orientation context for music creation, and discussed their rationale. The novice-orientation of CODES is characterized by a support for dynamic and prototypical music creation process and by actual cooperation, social knowledge construction, argumentation and negotiation among the different actors of musical prototypes design activities mechanisms. It enables knowledge sharing by means of rich (even being novice-oriented) interaction and adapted to address the idiosyncrasies of this CMP context.

[1] http://www.useit.com/papers/heuristic/heuristic_list.html

CODES has shown that Web-based networked music environments can offer even more than "consumer" possibilities for novices in music. Since we have integrated adequated tools, processes, and concepts in one single environment, novice users can create music prototypes, cooperate effectively and experience the feeling of being the creators of their own music. Music creation by novices is ultimately about people having fun and entertainment (and maybe also learning), not about following a fixed set of rules for music composition. It is not also a matter of composing a song from the beginning to the end (such as linear music) but of creating an own sound sequences (non linear music).

Having access to the rich interaction and argumentation mechanisms in CODES, and experiencing the process of music prototyping, we believe users may get a better understanding of the complex activities which are musical creation and experimentation. In CODES, partners cooperate not only by means of explicit actions on a shared objects space and explicit conversation, but also by interpreting the actions and, above all, the comments of other actors in their creative process.

However, CODES is not just about supporting novice people: features built for novices help everyone whose musical skills are less than a musician's capability. If we think musical skills are continuum - people do not merely know or not know music - CODES is open and accessible to all of us, from ordinary users to musicians. Thus, if actual novices can learn a lot using CODES, musicians may be "novices" when using CODES as well, experimenting (new) ideas and changing opinions.

References

Cycling74 (2009). Max/msp. Available at: ¡http://www.cycling74.com¿. Access in: 02 mar. 2009.

Google (2009). Youtube. Available at http://www.youtube.com/. Access in: 02 mar. 2009.

Gurevich, M. (2006). Jamspace: Designing a collaborative networked music space for novices. In *NIME*, pages 118–123.

Hoppe, A. F., Miletto, E. M., Pimenta, M. S., and Flores, L. V. (2009). Cooperation in musical prototypes design. In *13th IEEE International Conference on Computer Supported Cooperative Work in Design*. IEEE.

Iazzetta, F. and Kon, F. (1998). Internet music: Dream or (virtual) reality. In *V Simpósio Brasileiro de Computação e Música*, pages 69–81, Belo Horizonte, Brasil.

Media, F. I. (2009). Myspace. Available at: http://www.myspace.com/. Access in: 02 apr. 2009.

Miletto, E. M., Pimenta, M. S., Hoppe, A. F., and Flores, L. V. (2009). Who are the web composers? In *HCI International*, Lecture Notes in Computer Science. Springer.

Nielsen, J. (1992). *Evaluating the thinking-aloud technique for use by computer scientists*, pages 69–82. Ablex Publishing Corp., Norwood, NJ, USA.

Rubin, J. (1994). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. Wiley.

Shum, S. B. (1996). *Design argumentation as design rationale*, volume 35. Marcel Dekker Inc, New York, the encyclopedia of computer science and technology edition.

Weinberg, G. (2002). The aesthetics, history, and future challenges of interconnected music networks. In *Internation Computer Music Conference*, Göteborg, Sweden. ICMA.

Yahoo (2009). Flickr. Available at: http://www.flickr.com. Access in: 02 mar. 2009.

Wind instruments synthesis toolbox for generation of music audio signals with labeled partials

Martín Rocamora[12] **, Ernesto López**[12] **, Luis Jure**[1]

[1]Estudio de Música Electroacústica – Escuela Universitaria de Música
Av. 18 de Julio 1772 – (598) (2) 403 64 44

[2]Instituto de Ingeniería Eléctrica – Facultad de Ingeniería
Julio Herrera y Reissig 565 – (598) (2) 711 09 74

Universidad de la República, Montevideo, Uruguay

rocamora{@eumus.edu.uy,@fing.edu.uy}, elopez@fing.edu.uy, lj@eumus.edu.uy

Abstract. *In this work a methodology is proposed and a set of software tools is released for the automatic generation of synthesized audio files accompanied with labels that describe the temporal evolution of the amplitude and frequency of each one of the partials present. The approach is to synthesize wind instruments sounds using a simple yet effective additive synthesis model based on [Horner and Ayers, 1998]. Some improvements over the original model are implemented and others suggested for future work. In the context of automatic extraction of musical content from audio, this data can be used as ground truth labels for the development and evaluation of algorithms intended for example to estimate partials parameters or track its evolution. This seems an interesting contribution, since manual annotation is a very time consuming task in this situation and a resource of this kind is not available for researchers at present.*

1. Introduction

When developing an algorithm for the automatic extraction of musical content from audio recordings (known as Music Information Retrieval, MIR) it is important to have labeled examples in order to train machine learning schemes, adjust parameters or perform systematic evaluations comparing the results with the annotations. For the learning to be effective and for the evaluations to be representative it is desirable to have a large number of annotated examples at disposal. Depending on the MIR problem considered, the labeled data needed may be different. For instance, the multiple fundamental frequency problem requires the fundamental frequency value of each audio source at a certain time interval.

Usually labels are produced from music recordings by experts that manually annotate the information of interest. Unfortunately, this manual process is in most cases an arduous and time consuming task. For this reason, available resources are quite limited. Among them, the Popular Music Royalty Free Music Database [Goto, 2006] and the data provided by the Music Information Retrieval Evaluation eXchange, MIREX [Downie, 2006], are examples of the most commonly used. Additionally, ambiguities may arise during manual annotation, in which subjective judgements and personal assessment take part [Yeh et al., 2007]. Recently semi-automatic means of labeled data generation were devised, in order to avoid the disadvantages of manual labeling.

An option is to record the live performance of a musical piece whose symbolic representation is available, for instance as a MIDI file. Ideally the MIDI sequence would be an appropriate description of the musical content of the recording. However, in practice differences between performance and symbolic representation do exist, for example due to variations in duration and time onset of notes. Thus, a precise time alignment is needed to adjust the labeling to the performance. This complex problem can be tackled by means of techniques such as dynamic programming [Ellis, 2008]. In addition, other kinds of differences can also exist (e.g. substitutions, deletions or additions of notes), that may require more elaborated solutions.

Another option to perform automatic labeling is to synthesize music from MIDI sequences. There are several ways of generating audio from a MIDI file. MIDI sequencers can be used with sound modules, synthesis software or audio samples of musical instruments. The main drawback of this method is the lack of realism or naturalness compared to real recordings. It is important to notice that MIDI information is sometimes inadequate for labeling. For example, the MIDI file does not describe the temporal evolution of the fundamental frequency during a note. Moreover, recorded samples may not be played in tune. In this case an audio file is usually created for each single track and the labels are obtained by means of monophonic fundamental frequency estimation techniques, as described in [Yeh et al., 2007].

Sometimes an MIR algorithm tries to follow the course of each partial of the sounds in a given recording, for example to identify and segregate the different sound sources or to detect the multiple fundamental frequencies. Manually generating labels for this purpose is an overwhelming task, and to the best of our knowledge there are no resources that provide such data. In this work a methodology is proposed and a set of software tools is released to automatically produce labels in this situation. The approach is to synthesize wind instruments sounds using a simple yet effective additive synthesis model based on [Horner and Ayers, 1998] that generates a dynamic spectrum. Although the synthesis model is not able to reproduce many of the nuances and particularities of wind instruments (e.g some articulations, noisy attacks, blowing noise and legato notes), it generates a wide range of timbres that are clearly identifiable and enables to precisely track the temporal evolution of the amplitude and frequency of each partial. In this way a polyphonic labeled database can be generated by synthesizing MIDI files with this toolbox, which is the main contribution of this work. Some improvements over the original synthesis model are implemented and others suggested for future work.

2. Synthesis of musical instruments

2.1. Timbre

In order to effectively synthesize the sound of a given musical instrument we should be able to recreate its *timbre*. This broad characteristic carries information about the source (such as material, shape, size), type of excitation, etc. Timbre perception is therefore a complex phenomena related to several physical properties. Classical theory of timbre [Helmholtz, 1954] considered the main features to be the waveform amplitude envelope and the spectral magnitude envelope. The former reveals characteristics about the oscillation type (e.g. dumped, forced) and the kind of driving force (e.g. impulsive, continuous). The latter describes how the energy is distributed along the frequency domain. The acoustic system of most musical instruments consists of an excitation source (e.g. vocal folds) and a resonator (e.g. vocal tract). Resonant frequencies, which depend on the size, shape and material of the resonator, emphasize certain spectral regions thus amplifying some harmonics. Therefore the spectral envelope of an instrument typically exhibits characteristic peaks, named *formants*, located at certain frequency regions. Sounds produced by the same instrument (similar timbre) at different pitches, have distinct amplitude relations among its frequency components (see figure 1).

Modern studies of timbre [Risset and Wessel, 1982] showed that during the course of a sound its spectrum changes dramatically (see figure 2). Amplitude and frequency of each partial varies with time and this behavior plays an important role in timbre perception. For this reason, a sound synthesis algorithm must be able to produce a dynamic spectrum.

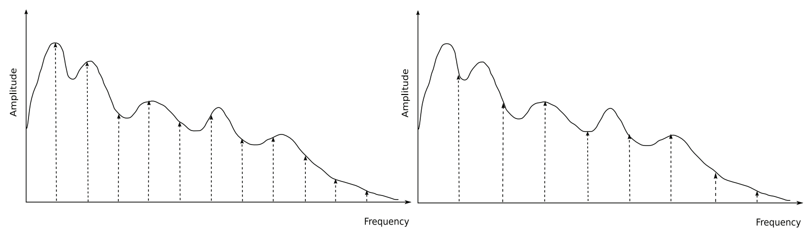


Figure 1: Diagram of the spectrum of harmonic sounds of similar timbre and different fundamental frequency (F0). Transfer function of the resonator exhibits peaks at certain regions. At different F0 the relative amplitude of frequency components is different.

2.2. Additive synthesis by analysis

A traditional sound synthesis technique, named *additive synthesis*, consists in the superposition of sinusoidal components whose frequency and amplitude typically vary with time producing a dynamic spectrum. In order to synthesize a given musical instrument sound, amplitude and frequency of a set of sound oscillators can be controlled with the information obtained by the

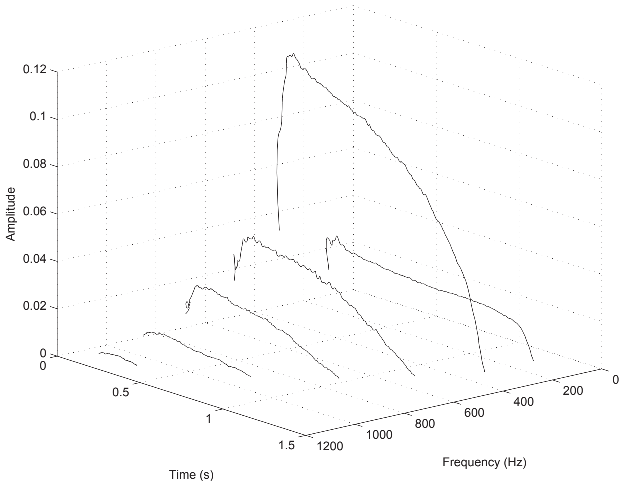


Figure 2: Temporal evolution of the first 6 partials of the sound of a French horn note (G3). Note that the amplitude evolution of each partial is independent. The analysis was performed using SMS software [Serra and Smith, 1990].

analysis of a real sound (like the one shown in figure 2). In this respect, the precise evolution of frequency and amplitude of partials is less important than the global or average behavior [Moore, 1990]. In particular, it is possible to build synthetic sounds that are perceived to be virtually identical to the original recordings, approximating the evolution of partial parameters with line segments or piecewise curves [Dodge and Jerse, 1997] (see figure 3).

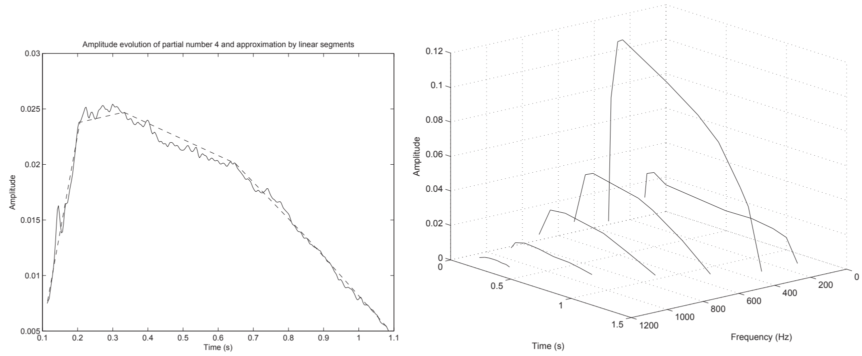


Figure 3: Partials amplitude evolution of the sound of figure 2 approximated by linear segments.

It is important to note that the analysis data is valid only within a small range of frequency and amplitude. This implies that the analysis of a certain note is not generally effective for synthesizing other pitches or dynamics of the same instrument. Changing the fundamental frequency of the note in the synthesis may probably not evoke the desired timbre. This is because in real instruments the location of formants remains the same, so relative amplitudes given by the analysis are changed for a different pitch (figure 1). The relative amplitudes of sound components also varies for different dynamics. Louder notes tend to increase the relative amplitudes of higher partials, making the slope of the spectral envelope less pronounced (see figure 4).

3. Wind instruments sound synthesis

3.1. Wind instruments sound characteristics

Beyond their evident singularities and differences, wind instruments have some common characteristics that motivate the idea of a general synthesis model as proposed in [Horner and Ayers, 1998]. Firstly, all of them consist of a tube resonator and some kind of acoustical excitation that makes the air within the tube to vibrate and resonate at a certain fundamental frequency. The excitation can be generated by the vibration of the performer's lips as in brass instruments (e.g. horn, trumpet, tuba, trombone), or by the vibration of a single or double reed as in woodwind instruments (e.g. clarinet, oboe, fagot, saxophone). In the case of flute type instruments (e.g. transverse flutes, recorders, organ flue pipes) the excitation comes from the effect

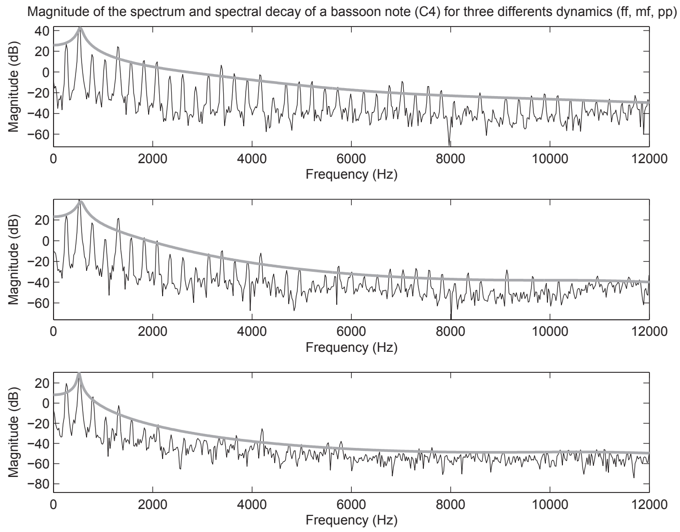


Figure 4: Magnitude of the spectrum and spectral decay of a bassoon note (C4) for three different dynamics (*ff*, *mf*, *pp*). The slope of the spectral decay gets steeper as intensity decreases.

of an air jet striking a sharp edge (edgetone). In either case, the vibration of the air column produces a harmonic spectrum whose components tend to decrease in amplitude with frequency. This spectrum is emphasized in some spectral regions given by the shape of the instrument (e.g. the bell). In this way, wind instruments exhibit characteristic formants [Hall, 2001]. See figure 5 for an example of the spectrum of oboe and clarinet.

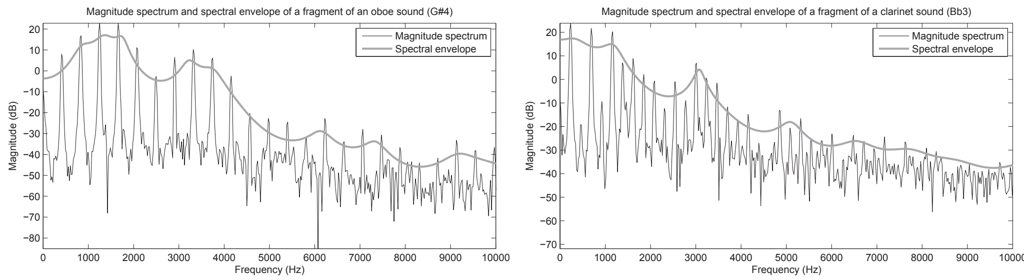


Figure 5: Spectrum of the sound of an oboe note (G#4) and a clarinet note (Bb3). Formants of the oboe give its *nasal* characteristic sound. Regarding the clarinet, its spectrum also exhibits formants and it is noticeable the absence (or relative weakness) of the second and fourth harmonic, typical of this instrument in the lower register [Fletcher and Rossing, 2008].

With regards to the waveform amplitude evolution, it can be usually divided into attack, steady-state and release. During attack and release the envelope can be roughly approximated with an exponential curve (see figure 6). However, some notes may have a more complex behavior, for example a pronounced attack followed by a short decay, steady-state and release.

Wind instruments also exhibit a broadly similar spectral behavior during the course of a note. Considering in turn each of the harmonics from the fundamental frequency, the attack tends to be slower and the release faster (see figure 6), so the harmonics seem to appear one after the other and fade out in the opposite way. Therefore, the sound gradually becomes brighter during the attack, until it reaches its maximum in the steady-state and it gets darker during the release. Additionally, brightness also changes with dynamics in wind instruments as described previously, that is, the relative amplitudes of higher harmonics increase with intensity [Fletcher and Rossing, 2008].

3.2. Contiguous partials grouping synthesis

The synthesis model adopted in this work is the one proposed in [Horner and Ayers, 2000] for the synthesis of the French horn and in [Horner and Ayers, 1998][Horner and Ayers, 2002] for the synthesis of wind instruments, using the *Csound* synthesis language. The idea is to apply an additive synthesis by analysis model with some simplifications. The model is described in the following for the sake of clarity. Firstly, contiguous partials are grouped in disjoint sets in order

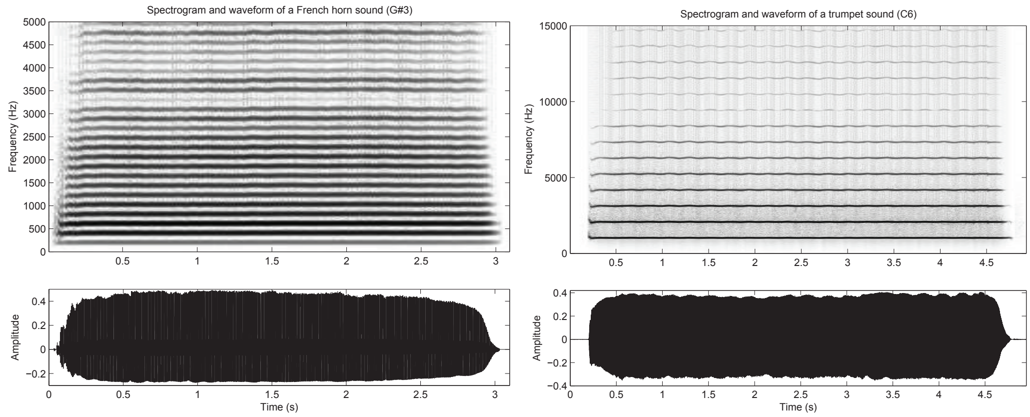


Figure 6: Spectrogram and waveform of the sound of a French horn note (G#3) and a trumpet note (C6). It can be noticed how the harmonics gradually appear one after the other during the attack and fade out in the opposite way during the realease. The waveform amplitude envelope can be approximated with an exponential curve during the attack and release.

to control their temporal amplitude evolution, following a perceptually motivated rule approximately corresponding to the division of the frequency range by critical bands. The first group has only the fundamental frequency component, the second group the second and third partials, the third group from fourth to seventh component and the fourth group has all the remaining harmonics. In this way, the amplitude evolution of each partial, rather than being independent, is such that components of a same group evolve in the same way. Another simplification consists in selecting a single representative spectrum of the steady-state. This spectrum is dynamically modified by changing the amplitude of each group in such a way to produce the behavior of the wind instruments described previously. The selection of the representative spectrum can be performed by means of optimization techniques, but in [Horner and Ayers, 1998] authors point out that picking a spectrum with average brightness gives a similar solution and requires much less computation. Both simplifications significantly reduce the amount of information that is retained from the analysis, by introducing an approximate and less precise model of the time evolution of the partials.

The block diagram of the synthesis model proposed in [Horner and Ayers, 1998] is depicted in Figure 7. The representative spectrum is divided into disjoint sets of partials and the waveform of each group is synthesized. Temporal amplitude evolution of groups of partials are controlled by a series of curves. For the first partial a curve $amp1$ consisting of linear segments is used. The other amplitude envelopes are exponentially related to the first one ($amp2 = amp1^2$, $amp3 = amp1^3$, $amp4 = amp1^4$). This relations assures that higher harmonics attack more slowly and decay earlier, as it is desired for wind instruments. Figure 8 shows an example of a six linear segments envelope for a neutral articulation. The authors propose other envelopes for different types of articulations, such as crescendo, decrescendo and forte-piano. The model also offers the possibility to determine the attack and decay times, which allows a better control of the articulation. The synthesized trumpet note of figure 8 shows the described spectral behavior. In addition, although the original amplitude envelopes are formed by linear segments, the exponential relation of the curves produces a global waveform amplitude envelope that is not linear but approximately exponential during attack and release.

Spectral differences along the instrument register are handled by using in many cases as few as two notes per octave as spectral reference. For a given pitch, the closest spectrum is synthesized, using the amplitude relations among its partials and the amplitude envelopes described above, but with the corresponding fundamental frequency. This approach has the drawback that it modifies the location of formants for pitches that do not match the reference spectrum notes and produces audible discrete timbre jumps along the instrument register.

The synthesis model also includes a dynamic vibrato that makes the instrument sound more natural, as performers typically modify their vibrato during the course of a note. Additionally, the synchronized frequency fluctuation of partials contribute to their perceptual fusion into a single tone [Chowning, 1999]. Vibrato rate changes from approximately 3 to 5 Hz, and a certain random perturbation is added to the starting and ending value so the vibrato is slightly different for each note. Also vibrato depth changes along the note as it is shown in figure 9, and its maximum value is controlled from the synthesis score program.

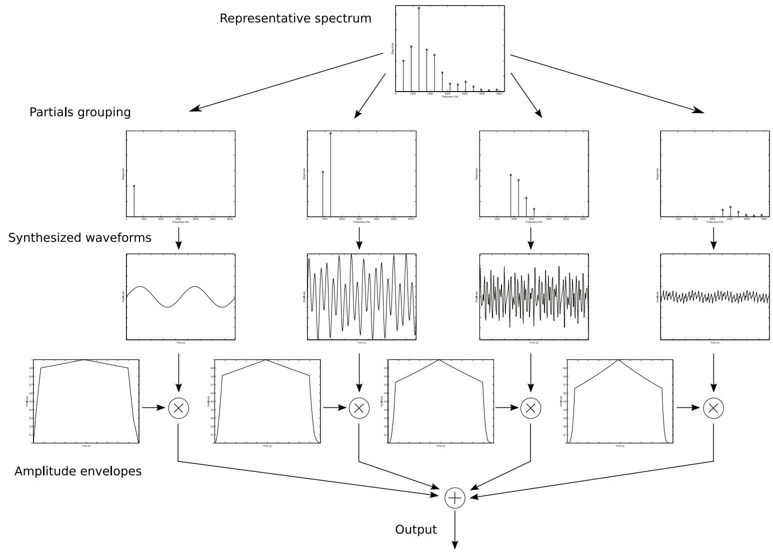


Figure 7: Block diagram of the synthesis system proposed in [Horner and Ayers, 1998].

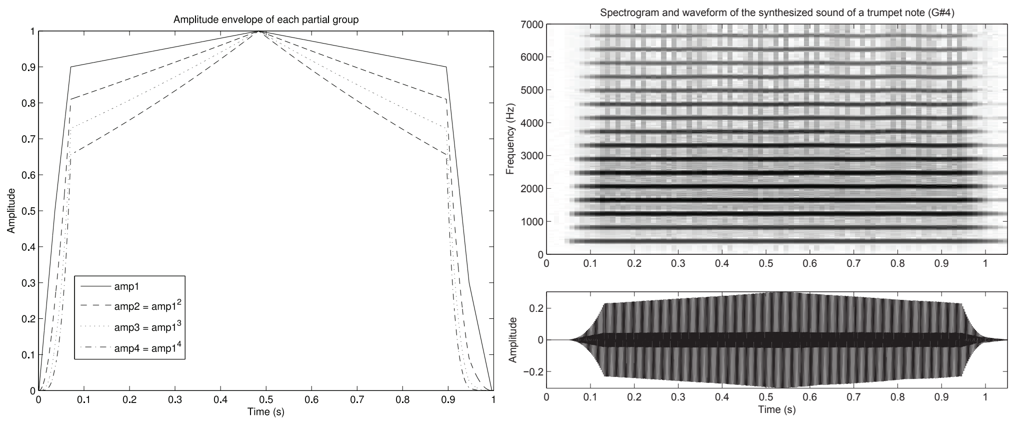


Figure 8: Curves used as amplitude envelopes for each group of partials and synthesized trumpet note. Successive harmonics attack more slowly and decay earlier. The resulting waveform amplitude envelope has an approximately exponential behavior during the attack and release.

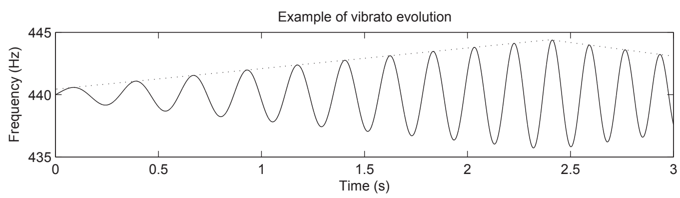


Figure 9: Vibrato evolution along the course of a note. Both vibrato rate and depth are variable.

3.3. Improvements over the original synthesis model

When aurally evaluating the output of the model, the synthetic nature of the sounds produced is recognizable. This is mainly due to lack of natural articulations and discrete timbre variations along the register. An additional deficiency of the model is that timbre changes produced by different dynamics are not taken into account. Therefore, increasing the number of spectral reference notes, including also different dynamics, would seem to offer important improvement. This could be achieved by analysing collections of musical instrument sound samples, such as McGill University Master Samples (MUMS)[1], Iowa Musical Instrument Samples (MIS)[2] and RWC Musical Instrument Sound Database[3]. Ideally, one reference spectrum should be available for each

[1] http://www.music.mcgill.ca/resources/mums/html/MUMS_audio.html
[2] http://theremin.music.uiowa.edu/MIS.html
[3] http://staff.aist.go.jp/m.goto/RWC-MDB/rwc-mdb-i.html

note and dynamic. This issue was explored in two ways in the present work. Firstly by using the SHARC Timbre Database[4] [Sandell, 1991], a collection of steady-state spectrum estimates obtained from the MUMS. Unfortunately, notes provided in MUMS cover only a single dynamic performance. For this reason, an automatic steady-state spectrum estimation procedure was implemented to analyse the MIS database, that includes notes played *ff*, *mf* and *pp*.

A straightforward way to assess the perceptual impact of the synthesis performed with a different reference spectrum for each note is to use the SHARC database. For a given note, it provides the amplitude and phase of the harmonics, corresponding to the sustain or steady-state portion of the tone. Using the SHARC data, several music fragments and chromatic scales were synthesized comprising various wind instruments. Aural tests showed smooth timbre variations of an instrument along its register, substantially improving the results of the synthesis performed with the original model data. During this process some unpleasant notes were identified, that could arise from spectrum estimation errors or erratic labeling and tuning problems of the MUMS database [Eerola and Ferrer, 2008].

In order to include different dynamics into the synthesis model, a steady-state spectrum database was built from the automatic analysis of MIS samples, following a procedure similar to the one applied in SHARC. It consists in selecting a representative spectrum of the sustain portion of the note and estimating the amplitude of the harmonics at this point. This process is depicted in figure 10. The steady-state portion of the note is considered as the longest time interval where the signal energy is greater than 75% of its maximum value. Then the spectrum of each steady-state signal frame is computed and they are summed up in an average spectrum. A representative time instant of the steady-state is determined from the frame whose spectrum most closely resembles the average spectrum, in a least-squares sense. Fundamental frequency is estimated at this time instant based on the autocorrelation function. The actual fundamental frequency may be different from the frequency of the note, due to tuning inaccuracies. After that, the Discrete Fourier Transform (DFT) of a four period length hann-windowed signal frame is calculated. Assuming that the signal is perfectly harmonic and stationary within the frame, partial amplitudes are picked from every fourth bin of the DFT up to 10 kHz.

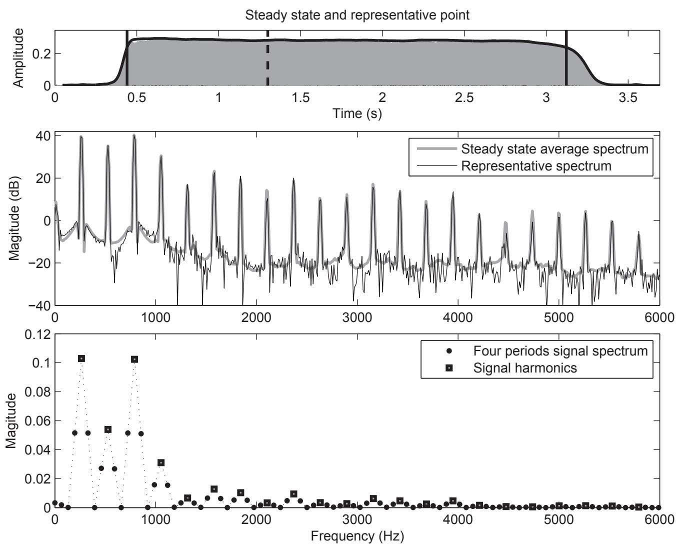


Figure 10: Estimation of a representative spectrum.

When applying this procedure in practice several difficulties were encountered. The criteria used to select the steady-state portion of the note fails for example if the energy exhibits a highly pronounced peak. This may happen in a note with a strong attack, as shown in figure 11. Some simple rules were added to the estimation process trying to avoid this kind of errors. In other cases it is even difficult to establish if a steady-state exists for a certain note. It is reasonable to suppose that some of the bad notes spotted when using SHARC data may come from these problems.

The database built in this form provides the smooth timbre variation previously described for the SHARC data (see figure 12), as well as simulates the natural variation of timbre for

[4] `http://www.timbre.ws/sharc/`

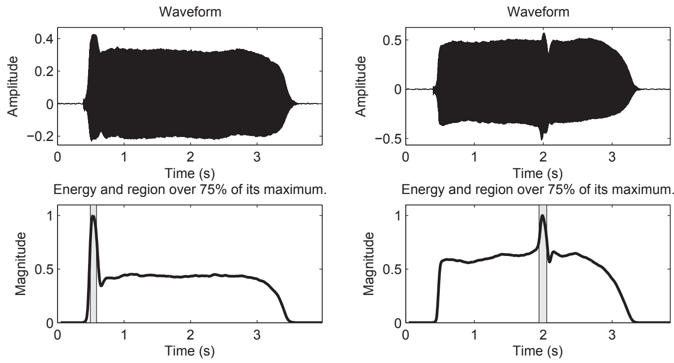


Figure 11: Examples of problems encountered during the spectrum estimation.

different dynamics (see figure 13). Due to the problems related to estimating a single representative spectrum for the whole note, the database inevitably contains some unsatisfactory notes. For this reason, it is necessary to aurally evaluate the synthesis for each instrument trying to identify these note in order to correct them.

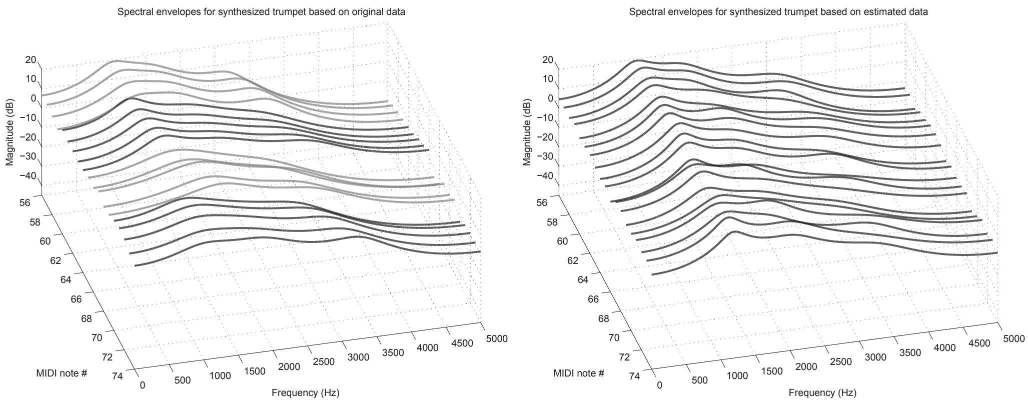


Figure 12: Spectral envelopes for the trumpet synthesized using the original and the estimated spectral data. In the first case, groups of contiguous notes are clearly noticed. Due to this approximation, location of formants is changed from one note to the other. The estimated spectral data presents a much more regular behavior of the spectral envelopes.

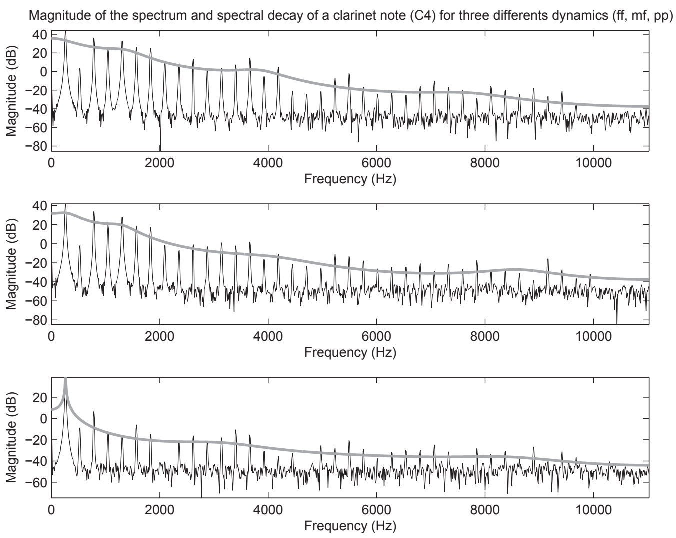


Figure 13: Magnitude of the spectrum and spectral decay of a synthesized clarinet note (C4) for three different dynamics (*ff*, *mf*, *pp*). The spectral decay shows a similar behavior to that of real wind instrument sounds.

3.4. Implementation and examples

The synthesis model was implemented as a publicly available[5] toolbox library in *Matlab / GNU Octave*, based on the *Csound* code in [Horner and Ayers, 2002]. The analysis data used can be chosen from the original data, the SHARC database and the data estimated from MIS audio samples. Figure 14 shows a dependency graph of the functions in the library.

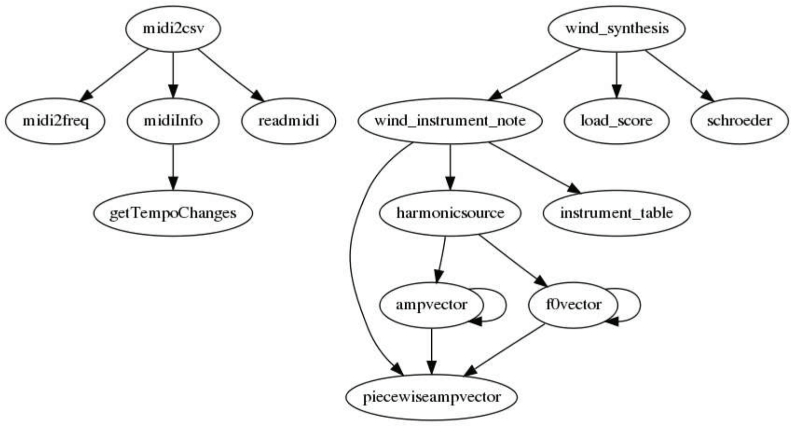


Figure 14: Dependency graph of the functions in the library.

The code in table 1 is an example of usage. The main function is `wind_synthesis`, which returns a vector (`d`) and an array of structures (`note`). The former corresponds to the synthesized audio signal samples, that are also saved to an audio file. The latter represents the data of each note: the name of the instrument, the value of frequency and amplitude of each partial at sample time instants given in a vector of temporal indexes. The function has a set of input arguments that take default values if they are not specified. The first argument is sampling frequency (`fs`), followed by the name of the score input file (`in`) and the name of the output audio file (`out`). It can also be specified if reverberation is added to the synthesis (`reverb`), and if so, also the value of reverberation time T60 (`delay`) and the percentage of reverberated signal added to the output (`pr`). The reverberation uses an implementation of a Schroeder reverberator (4 comb filters followed by 2 all-pass filters) available at [Väänänen, 2000]. Finally an argument controls output verbosity (`verbose`) and another one sets the spectrum data used in the synthesis (`tables`).

```
%% wind synthesis example program
% global parameters
fs = 44100;
verbose = 0;
% reverb paremeters
reverb = 1;    % whether to use reverb or not
delay  = 1.2;  % delay time
pr     = 0.25; % percent of reverberated signal
% output filename prefix
out = 'wind_synthesis-test';
% synthesis tables to use (0: Horner and Ayers - default, 1: MUMS Sharc, 2: MIS Iowa)
tables = 0;
%% score file
in ='scores/Beethoven_Op18-5-var5_quinteto.csv';
% 2 piccolo flutes, 2 oboes, 2 clarinets, 2 bassoons
%% synthesis
[d notes] = wind_synthesis(fs, in, out, reverb, delay, pr, verbose, tables);
```

Table 1: Example code of the library usage.

The score is loaded from a comma separated values file as shown in table 2. If the first row starts with a 0, the second parameter is used to set the beats per minute (bpm, default to 60). Each of the remainder rows correspond to a note. The first parameter sets the instrument number, from the ten wind instruments available (horn, clarinet, oboe, bassoon, flute, piccolo, sax, trumpet, tuba and trombone). Then onset time and duration are specified in beats (or seconds if no bpm is set). The following parameters are the amplitude (a value between 0 and full scale, $2^{16}/2$), the frequency in hertz and the vibrato depth (a value between 0 and 1). Finally the attack and decay time are specified in seconds. It is also possible to perform the synthesis from a MIDI

[5] `http://iie.fing.edu.uy/~rocamora/wind_synthesis/doc/`. Some audio examples are also available.

file. For this purpose a set of functions are provided (adapted from [Schutte, 2008]) to convert a MIDI file into the appropriate comma separated values file.

INSTRUMENT	START	DURATION	AMPLITUDE	FREQUENCY	VIBRATO	ATTACK	DECAY
0,	118,	0,	0,	0,	0,	0,	0
2,	1.500,	0.650,	6000,	294.300,	0.000,	0.050,	0.100
2,	1.950,	0.700,	6500,	490.500,	0.000,	0.050,	0.100
2,	2.450,	0.750,	7000,	392.400,	0.000,	0.050,	0.100
1,	3.100,	10.000,	4000,	294.300,	0.000,	0.060,	0.250
1,	3.000,	10.000,	4000,	196.200,	0.000,	0.060,	0.250
2,	3.100,	2.150,	7500,	588.600,	0.000,	0.050,	0.100
2,	5.150,	0.200,	7000,	654.000,	0.000,	0.030,	0.030

Table 2: Score fragment for clarinet and French horn. First row sets the bpm value.

The library also includes score examples that consider each instrument individually and several polyphonies. Some of them were adapted from [Horner and Ayers, 2002], others come from MIDI files available at *Mutopia* (`http://www.MutopiaProject.org`) under Creative Commons licence and the remaining were prepared for the library from different sources. Figure 15 shows the graphical output of the toolbox for the synthesis of a French horn phrase and a fragment of a piece for oboe, clarinet and bassoon.

4. Conclusions and future work

This work suggests a methodology for creating music audio examples accompanied by labels that describe the evolution of amplitude and frequency of each one of the partials of the sounds present. The method is based on the synthesis of wind instrument sounds following an additive synthesis model described in [Horner and Ayers, 1998]. Despite the aforementioned limitations, the synthesis model is effective since it is possible to clearly identify the wind instrument being synthesized and it also capable of generating a wide range of timbres. Besides, it enables to precisely track the temporal evolution of the partials at low computational cost. Noticeable improvements over the original model, regarding the naturalness of the synthesis, were obtained by automatically analysing the MIS musical instrument sound samples collection in order to gather spectral information of each note played at different dynamics for several wind instruments. In addition a set of software tools is released, that given its ability to synthesize MIDI files, is an useful tool to automatically build a labeled audio database with annotated amplitude and frequency evolution of each partial. This seems and interesting contribution, since manually generating labels for this purpose is an overwhelming task and to the best of our knowledge no such data is available. The methodology of using group additive synthesis for the automatic generation of labeled partials can be extended to other families of timbre [Lee and Horner, 1999], what would provide a richer annotated database. We have used the library in our research and it has proven to be a very handy tool in the development of algorithms for certain MIR applications.

Future work will include the improvement of the spectral data available, aiming a complete chromatic scale along the whole register for each instrument with different dynamics for every note. At present, these conditions are not completely fulfilled because some notes and dynamics are missing in the MIS database and also estimation errors can exist. The current implementation only includes the amplitude envelope illustrated in figure 8. Improvement of the synthesis algorithm would include the implementation of the other amplitude envelopes proposed for the original model. The spectral estimation procedure as well as the synthesis model assume perfect harmonicity of the spectrum. Further research involves the study of the degree of inharmonicity of the available sound samples and the assessment of its eventual impact in the synthesis, evaluating the pertinence of its inclusion in the model. Another interesting area for future research is the addition of an effective noise model to mimic the blowing noise that is clearly perceived in some soft music passages or closely miked solos.

Optimal synthesis results require a carefully prepared score with a detailed control of the amplitude and temporal location of each note, as well as the type of envelope used and the attack and decay times. Even though a well sequenced MIDI file will include the appropriate information about time and amplitude (in the form of velocity) of the notes, the rest of the nuances of articulations and phrasing will be missing. A desired goal is to produce an ample corpus of high quality scores for the toolbox illustrating a variety of musical situations. Finally, the improvement of the toolbox usability and flexibility as well as its implementation as a standalone application is being considered.

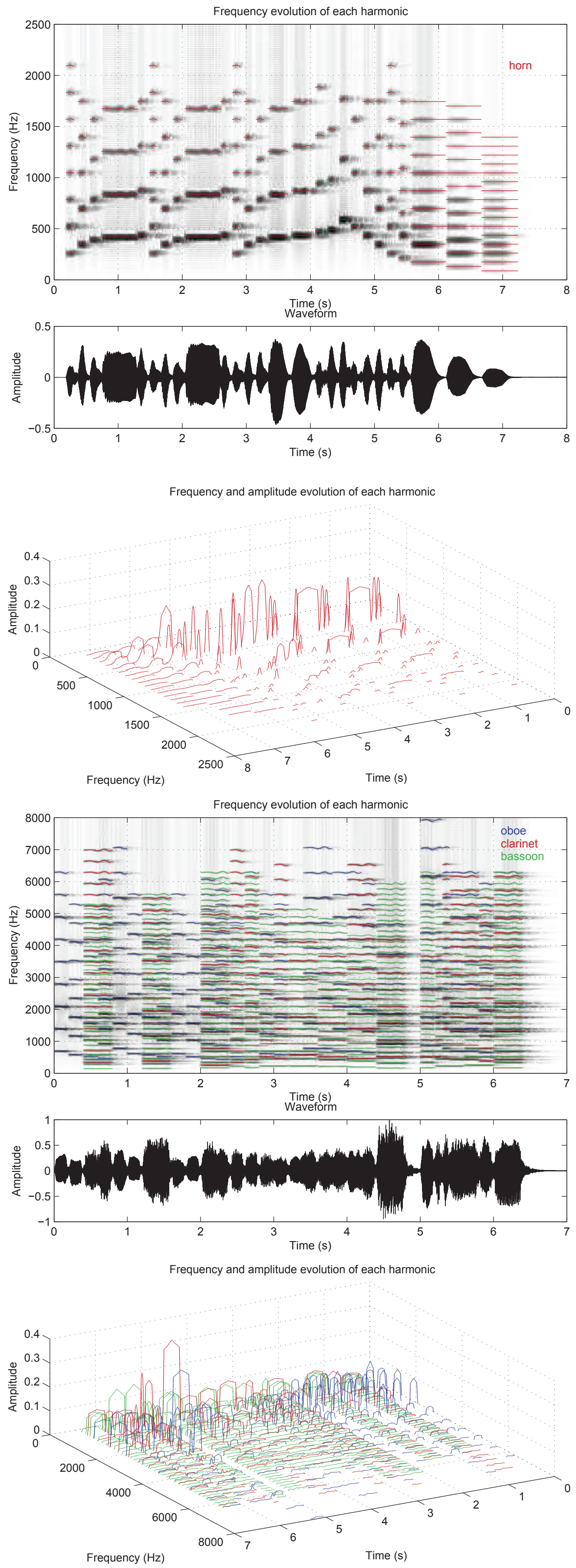


Figure 15: Graphical output of the toolbox for two of the included examples, a French horn phrase and a fragment of a piece for oboe, clarinet and bassoon.

References

Chowning, J. (1999). Perceptual Fusion and Auditory Perspective. In Cook, P. C., editor, *Music, Cognition, and Computerized Sound*. The MIT Press.

Dodge, C. and Jerse, T. A. (1997). *Computer Music: Synthesis, Composition, and Performance*. Schirmer, 2 edition.

Downie, J. S. (2006). *The Music Information Retrieval Evaluation eXchange (MIREX)*. D-Lib Magazine 12 (12). Available from `http://www.dlib.org/dlib/december06/downie/12downie.html`.

Eerola, T. and Ferrer, R. (2008). Instrument Library (MUMS) Revised. *Music Perception*, 25(3):253–255.

Ellis, D. P. W. (2008). "Aligning MIDI scores to music audio", web resource. Available from `http://www.ee.columbia.edu/~dpwe/resources/matlab/alignmidiwav/`. Accessed on 9 August 2009.

Fletcher, N. H. and Rossing, T. D. (2008). *The Physics of Musical Instruments*. Springer, 2nd edition.

Goto, M. (2006). AIST Annotation for the RWC Music Database. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, pages 359–360. Available from `http://staff.aist.go.jp/m.goto/RWC-MDB/AIST-Annotation/`.

Hall, D. E. (2001). *Musical Acoustics*. Brooks Cole; 3rd Revised edition.

Helmholtz, H. (1954). *On the Sensations of Tone*. Dover Publications.

Horner, A. and Ayers, L. (1998). Modeling Acoustic Wind Instruments with Contiguous Group Synthesis. *Journal of the Audio Engineering Society*, 46(10):868–879.

Horner, A. and Ayers, L. (2000). Contiguous-Group Wavetable Synthesis of the French Horn in Csound. In Boulanger, R., editor, *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing and Programming*. The MIT Press.

Horner, A. and Ayers, L. (2002). *Cooking with Csound, Part 1: Woodwind and Brass Recipes*. A-R Editions.

Lee, K. and Horner, A. (1999). Modeling piano tones with group synthesis. *Journal of the Audio Engineering Society*, 47(3):101–111.

Moore, F. R. (1990). *Elements of Computer Music*. Prentice Hall.

Risset, J.-C. and Wessel, D. L. (1982). Exploration of timbre by analysis and synthesis. In Deutsch, D., editor, *The Psychology of Music*, pages 26–58. New York: Academic.

Sandell, G. J. (1991). A Library of Orchestal Instrument Spectra. In *Proceedings of the 1991 International Computer Music Conference*, pages 98–101, San Fransico, California.

Schutte, K. (2008). MIDI for Matlab. Available from `http://www.kenschutte.com/midi`.

Serra, X. and Smith, J. (1990). Spectral modeling synthesis a sound analysis/synthesis based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14:12–24. SMS, Available from `http://www.mtg.upf.edu/technologies/sms`.

Väänänen, R. (2000). Matlab implementation of a schroeder reverberator. Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing. Available from `http://www.acoustics.hut.fi/~riitta/.reverb/schroeder.m`.

Yeh, C., Bogaards, N., and Röbel, A. (2007). Synthesized polyphonic music database with verifiable ground truth for multiple F0 estimation. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*, pages 393–398. Available from `http://recherche.ircam.fr/equipes/analyse-synthese/cyeh/database.html`.

Descoberta Automática de Conhecimento em Interpretações Musicais: Microandamento e Microdinâmica

Fúlvio Figueirôa Silvestre , Raphael Freire de O. Holanda , Geber Lisboa Ramalho

Centro de Informática – Universidade Federal de Pernambuco
Caixa Postal 7851 – 50740-540 Recife, PE

{ffs, rfoh, glr}@cin.ufpe.br

Abstract. *It is known that music expressive performance is closely related to slight deviations in time and dynamics [Widmer e Goebl, 2004]. Although these deviations occur according to the musician interpretation knowledge, it is hard to capture this knowledge by musician verbalization. The "Um país um violão[1]" project aims to study the expressive music performance in Brazilian Popular Music, Bossa Nova Guitar in particular, by using computers to analyze performances of human musicians. Previous works were focused on automatically indentifying rhythm patterns [de Lima, 2007]. The work presented here analyzes patterns of Microtiming and Microdynamic in bossa nova guitar interpretation. The first results are promising, since they provide evidences of the discovery of novel ethnomusicological knowledge concerning bossa nova guitar.*

Resumo. *Sabe-se que a interpretação expressiva da música, t|raduz-se, entre outras, em pequenos desvios de tempo e dinâmica [Widmer e Goebl, 2004]. Embora, esses desvios ocorram de acordo com o conhecimento do interprete, é bastante difícil capturar este conhecimento interpretativo via verbalização do músico. O projeto "Um país um violão" tem como objetivo estudar a Expressividade Musical no âmbito da Música Popular Brasileira, em especial o violão de Bossa Nova. Trabalhos anteriores [de Lima, 2007] estavam focados em descobrir automaticamente padrões rítmicos. O trabalho aqui apresentado analisa os fenômenos do Microandamento e da Microdinâmica na interpretação violonística da bossa nova. Os primeiros resultados dão indícios da descoberta de novos conhecimentos etnomusicológicos acerca do violão de bossa nova.*

1. Introdução

Nos últimos anos, vem-se estudando, com o auxílio do computador, como um intérprete (músico) executa de forma expressiva uma determinada peça ou canção [Zanon e Widmer, 2003, Goebl et al., 2004, Widmer, 2001]. Sabe-se que a interpretação expressiva da música traduz-se, entre outras, em pequenos desvios de tempo e dinâmica que não estão explicitamente anotados em uma partitura, grade de acordes, tablatura, etc. Costuma-se dar o nome de *microtiming* [Gouyon, 2007, Wright e Berdahl, 2006] (que chamaremos de microandamento) os desvios no tempo, em que as notas são tocadas alguns milisegundos antes ou depois do previsto no andamento normal. Além desse fenômeno, estudaremos aqui o que chamamos de microdinâmica, a saber, os desvios não explicitamente anotados que ocorrem nas intensidades das notas tocadas.

Embora esses desvios ocorram de acordo com o conhecimento do intérprete, é bastante difícil capturar esse conhecimento interpretativo via verbalização, em um processo

[1]"a country, a guitar", a reference to "Corcovado" lyrics (by Tom Jobim)

clássico de aquisição de conhecimento. A descoberta desse conhecimento é importante por duas razões básicas: (a) a explicitação das "regras" de interpretação permite uma melhor formação dos músicos, em particular, nos estilos musicais em que a tradição oral prevalece, onde há pouca notação, como é o caso das grades de acordes ou assemelhados encontrados nos livros da música popular brasileira [Chediak, 1994]; (b) tais "regras" podem ser utilizadas para que um computador seja capaz de interpretar uma peça ou canção com expressividade similar à de um músico humano.

Nesse contexto, o computador torna-se um instrumento indispensável para analisar esse tipo de fenômeno. De fato, o computador pode identificar desvios dificilmente mensuráveis pelo ouvido humano e realizar uma análise mais detalhada e com uma objetividade que as pessoas, na maioria das vezes, não conseguem atingir.

Atualmente, existe uma área de estudos que cuida justamente dessas características, chamada de expressividade musical. Essa área vem sendo estudada por várias disciplinas, como a psicologia e a computação musical. Na psicologia, esse estudo se dá através de percepções do ouvinte e intenções do intérprete. Já na computação musical, isso ocorre por meio da análise minuciosa de logs de interpretações de intérpretes humanos.

A maior parte do trabalho atualmente feito em computação musical, nessa área de interpretação expressiva, tem como foco a música clássica e o piano [Zanon e Widmer, 2003]. Existem pouquíssimos [Gouyon, 2007] sobre expressividade no domínio da musica não-clássica e, praticamente, nada no caso do violão popular brasileiro, que é um dos seus principais ícones.

Trabalhos anteriores de nosso grupo de pesquisa no contexto do projeto "Um país um violão" [de Lima, 2007] estavam focados em descobrir automaticamente padrões rítmicos. O trabalho aqui apresentado analisa os fenômenos do microandamento e da microdinâmica na interpretação violonística da bossa nova. Os primeiros resultados dão indícios de novos conhecimentos etnomusicológicos sobre o violão de bossa nova.

2. Expressividade Musical

Em uma peça musical ou canção, características como altura, duração, variação no tempo, intensidade e timbre não fazem parte apenas da estrutura musical, mas demonstram também a intenção do intérprete e a forma subjetiva e pessoal de como cada músico executa uma partitura. O estudo da expressividade musical é de extrema importância para uma melhor análise dessas características. Esse estudo tem levado diversos pesquisadores, tanto no âmbito da computação musical como da psicologia, a descobrirem fatos relevantes, a partir dos quais, podemos utilizá-los para aprimorar, por exemplo, o ensino musical [Sundberg et al., 1991]. Com o advento dos computadores, foi possível realizar uma análise um tanto minuciosa, feita através de inúmeras pesquisas neste tema e com resultados bastante interessantes. Esse trabalho aborda duas importantes características da expressividade musical: microandamento (do inglês *microtiming*) e microdinâmica (do inglês *microdynamic*) no âmbito do violão popular e, mais especificamente, da bossa nova, tratando-se portanto, de um estudo inédito no ritmo citado.

Embora os primeiros estudos tenham sido no final do século XIX, somente a partir do século XX, com o surgimento de inovações tecnológicas - em especial a computação científica - foi permitida uma análise mais precisa e de um conjunto maior de dados, fazendo com que as pesquisas em expressividade musical produzissem melhores e mais confiáveis resultados.

Diante desses primeiros estudos, já foram constatadas importantes variações, não apenas no que diz respeito ao comportamento do instrumentista perante a obra que executa, mas também, frente aos mecanismos de percepção envolvidos na escuta.

Alguns importantes estudos sobre expressividade comprovam o que foi exposto anteriormente. Sundberg e seus colegas [Sundberg et al., 1991] buscaram identificar parâmetros acústicos envolvidos em uma performance musical, com a finalidade de quantificar as pequenas e grande variações de tempo, dinâmica, timbre e afinação. Essas variações formam a microestrutura de uma performance e diferenciam performances distintas da mesma partitura [Palmer, 1997]. Uma vez quantificadas essas variações, o passo seguinte seria entender onde reside o impacto emocional de uma execução e como esse impacto é conduzido.

Um importante grupo de pesquisa nessa área é o Instituto de Pesquisa *Österreichisches Forschungsinstitut für Artificial Intelligence* - ÖFAI - de Viena, liderado por Gehard Widmer, que desenvolveu um modelo baseado em técnicas de *machine learning* e *data mining*, para o reconhecimento automático de padrões de parâmetros descritores de expressividade musical em um grande volume de dados. O modelo reconheceu execuções dos artistas: Rubisntein, Maria João Pires, Horowitz e Maurizzio Pollini [Widmer et al., 2003], [Zanon e Widmer, 2003], [Goebl et al., 2004]. O modelo proposto se mostrou bastante eficaz na descrição e quantificação de uma performance de forma objetiva, tanto de músicos profissionais como estudantes.

O que objetivam esses grupos de pesquisas é basicamente responder as seguintes questões: existem princípios explicáveis e quantificáveis que governam a expressividade de uma performance?; Em que medidas e até que ponto são aceitáveis performances executadas em uma música?; Quais são os princípios cognitivos que governam a execução (no intérprete) e a percepção (no ouvinte) na expressividade musical?; E, por fim, como isso é feito e com qual experiência musical? [Widmer, 2001]. Nosso estudo tenta buscar respostas para as duas primeiras questões, particularmente tratando o ritmo da bossa nova nas dimensões de tempo e dinâmica.

Os primeiros estudos sobre dinâmica surgiram no final do século XIX. [Binet e Courtier, 1895] conseguiram registrar a força com que era pressionada a tecla de um piano. Para isso, utilizaram um pequeno tubo de borracha posicionado embaixo das teclas. À medida que essas teclas eram pressionadas, pulsos de ar formados pelo tubo controlavam uma agulha que registrava a ação em um papel em movimento. Com isso, foi possível investigar a execução de trinados, acentos e variações de dinâmica. Esse estudo possibilitou identificar padrões de ações conduzidas por pianistas para realizar gestos expressivos, como por exemplo, um acento, pois além de imprimir maior tensão na tecla acentuada, o intérprete toca a nota precedente mais destacada e a nota acentuada um pouco alongada e mais ligada à nota seguinte [Gabrielsson, 1999]. Em 1898, [Ebhardt, 1898] publicou um estudo onde ele utilizava dispositivos eletromecânicos para registrar o pressionamento das teclas do piano que assim como [Binet e Courtier, 1895], também identificou alongamentos em notas acentuadas.

Estudos sobre microandamento tiveram início no mesmo período que os estudos de microdinâmica, já que ambos são subconjuntos da expressividade musical, entretanto, ao longo destes anos a maioria dos pesquisadores sempre procurou dar uma maior ênfase a análise de microandamento, pois consideram os resultados mais significativos do ponto de vista do estudo da expressividade musical.

Microandamento deve consistir em pequenas, mas, significantes, variações do exato momento em que a nota deve ser executada, sem contudo, fazer com que a música perca sua corretude. Esses desvios são facilmente identificados por um leigo quando, por exemplo, um computador realiza uma performance com os tempos executados no momento exato (sem microdesvios). Nesse caso, percebe-se claramente a forma mecânica com que a máquina toca. Freqüentemente, esses microdesvios são classificados inapropriadamente como discrepância , ruídos ou imprecisões . No entanto, diversos estudos dedicam-se a cobrir essas, assim chamadas, “imprecisões”, as quais transmitem informações sobre a estrutura musical, mas também provêem uma janela sobre a representação cognitiva da música.

Sears, em 1902 [Sears, 1902], publicou um dos primeiros estudos em microandamento, onde, utilizando também dispositivos eletromecânicos, mediu variações na duração de notas de mesmo valor, na duração de compassos e nas proporções entre durações de notas de valores distintos tocados por organistas [Gabrielsson, 1999]. [Sundberg e Verrillo, 1980] propuseram que cada intérprete segmenta as frases de uma mesma partitura individualmente, delimitando o início e o final das mesmas a partir de desvios de tempo. Posteriormente, [Todd, 1985], propôs um modelo computacional para os desvios temporais que enfatizam a hierarquia das frases musicais. Esse modelo estabelece relações entre variações de tempo de performance e o comportamento de um corpo em movimento utilizando equações de cinemática [Todd, 1995]. Já [Clynes, 1995], formalizou padrões de variação de tempo relacionado a compositores específicos.

Dois estudos importantes foram publicados recentemente sobre o jazz: [Freeman e Lacey, 2002] identificaram janelas de 30 milissegundos ao redor de uma batida, enquanto que [Friberg e Sundström, 2002], caracterizaram um padrão longo/curto de colcheias que proporciona o swing do jazz.

Os estudos que analisamos aqui apontam para a existência de padrões de microandamento e microdinâmica na música clássica. Sendo assim, partimos do pressuposto que existem padrões também na bossa nova e a questão, então, passa a ser como e onde observar tais padrões.

Conforme observamos até o momento, a maioria dos estudos sobre microandamento e microdinâmica estão voltados para a música clássica e, principalmente, para o piano. No âmbito da música popular brasileira, não encontramos análises que fizessem alusão à microdinâmica. Entretanto, podemos citar dois importantes estudos de microandamento: Fabien Gouyon [Gouyon, 2007] estudou o microandamento em Samba de Roda, Matthew Wright e Edgar Berdahl [Wright e Berdahl, 2006] realizaram estudos percursivos em nove ritmos brasileiros. Importante observar que o foco desses estudos foi o ritmo, mas tocado por instrumentos percussivos.

A maioria dos trabalhos que estuda microandamento e microdinâmica se propõe a tentar descobrir características a cada unidade de tempo (semínima). Desta forma, podemos fazer o seguinte questionamento: dado que a bossa nova é composta de frases, onde cada frase possui dois compassos de dois tempos cada compasso, existe alguma semelhança entre a análise feita na semínima e a análise feita na frase? E no caso de observarmos também o compasso individualmente, existe alguma característica comum com as análises anteriores?

3. Método adotado

Na seção anterior, descrevemos o estado da arte sobre pesquisas em expressividade musical, onde observamos que existem importantes estudos neste sentido abordando os mais variados rítmos e instrumentos, assim como as mais variadas características. Entretanto, o rítmo (bossa nova) e o instrumento (violão) escolhidos por nós, tornam inéditos os estudos de microandamento e microdinâmica nesses aspectos de uma performance. A partir de agora, falaremos um pouco sobre o modelo por nós proposto para resolução do problema descrito acima.

3.1. Corpus Analisado

Para fazer a análise da expressividade musical, é importante, primeiramente, capturar os dados da canção a ser analisada. Há, basicamente, duas maneiras de realizar essa captura de dados: (a) diretamente do áudio, onde se tem a canção de forma fiel como foi executada, porém a extração de informação simbólica, mais facilmente manipulável pelo computador, pode ser bastante complexa; (b) por meio de instrumentos MIDI, que já fornecem uma saída simbólica (em formato MIDI), porém exige do pesquisador que ele disponha de tais instrumentos. Essa segunda forma foi a escolhida no projeto por já fornecer diretamente o que precisávamos e por já dispormos de um violão MIDI. De toda forma, junto com a captura MIDI, também foi gravado, ao mesmo tempo, o áudio dessas canções para uma eventual necessidade futura.

Havia, porém, dúvidas quanto a confiabilidade da captura via um violão MIDI, principalmente no que diz respeito à dimensão temporal, de cuja precisão depende a análise de microandamento. Foi realizado um estudo comparativo entre a detecção de ataques indicada pelo violão MIDI e algoritmos de detecção de ataque aplicado ao áudio gravado simultâneamente à captura MIDI [Júnior, 2006]. Os resultados mostraram que não havia diferença estatisticamente relevante entre os dois, concluindo que a informação de detecção de ataque do violão MIDI é tão confiável quanto o que se pode obter hoje a partir do áudio.

Para que as gravações ocorressem da forma mais natural possível, não foi utilizado nenhum tipo de metrônomo, deixando assim o intérprete livre quanto a sua expressividade. Em contrapartida, criou-se a necessidade de serem adicionados marcos a partir dos quais as análises deveriam ser feitas. Foi então, que [de Lima, 2007] inseriu os *beat trackings* ou pulsações. Essa estrutura métrica foi inserida nas obras baseada no aplicativo *BeatRoot*, criado por [Dixon, 2001], que é uma ferramenta de indução de pulsação e funciona de maneira interativa, mesmo em canções onde existem grandes e bruscas mudanças no andamento. Primeiro, o sistema induz as suas pulsações da obra em análise para, em seguida, o usuário poder ouvir os resultados, corrigindo eventuais imprecisões, como, por exemplo, pulsos que foram induzidos erroneamente.

Outra característica importante, que a base de dados possui, é que a mesma encontra-se livre de ruídos. Chamou-se ruídos, eventos estranhos que ocorreram nas canções, por exemplo, eventos onde o *velocity*[2] é desprezível, ou eventos impossíveis de acontecerem num violão devido à anatomia das mãos, entre outros. Esses eventos foram limpos primeiramente por [de Lima, 2007]. No entanto, ele encontrou alguns problemas, alguns dos quais foram corrigidos posteriormente por [Scholz, 2008]. Esse processo de limpeza de dados foi de extrema importância para nosso trabalho, já que, com os dados corrigidos, pôde-se voltar as atenções para parte de análise.

Saindo um pouco das características técnicas de armazenamento, e devido ao foco do projeto ser o estudo da bossa nova, os intérpretes selecionaram algumas canções de João Gilberto de seus respectivos repertórios e ficaram livres para executá-las de acordo com sua expressividade, seguindo apenas as cifras previamente fornecidas.

A Tabela 1 mostra em ordem alfabética quais canções foram gravadas e por quem:

[2]**Velocity**: é um atributo dos eventos MIDI que indica a intensidade com que um evento deve ser executado. Equivale a força com que a nota é tocada pelo intérprete nos arquivos capturados por violões MIDI.

Canções	Intérprete 1	Intérprete 2
A Felicidade	X	
Bim Bom		X
Chega de Saudade	X	
Corcovado	X	
Desafinado	X	
Eu Sei Que Vou Te Amar	X	
Garota de Ipanema	X	X
Insensatez	X	X
O Barquinho		X
Samba De Uma Nota Só	X	
Só Danço Samba		X
Tarde em Itapoã	X	
Wave	X	X

Tabela 1: Gravações

Com base na tabela acima, observa-se que foram gravadas canções diferentes pelos intérpretes, mas também, que a mesma canção foi gravada por ambos em três ocasiões: Garota de Ipanema, Insensatez e Wave. Isso foi feito propositalmente para aumentar a robustez da análise. Verificou-se, também, que foram gravadas dezesseis canções, com uma média de dois minutos por cada canção, o que nos dá um universo bastante razoável de dados a serem analisados.

3.2. Implementação

Nesta seção, iremos analisar a forma como foi codificada a solução do problema, demonstrando o algoritmo (Figura 1) que foi utilizado para realizar os estudos sobre microandamento e microdinâmica no âmbito da bossa nova.

```
para cada arquivo faça:
       obtenha os beat tracking //eventos da trilha 7
       obtenha os eventos midi das demais trilhas // trilhas 1..6
       para cada uma das trilhas faça:
              obtenha os elementos da trilha corrente
              para cada um dos beat tracking faça:
                     calcule o intervalo entre o beat tracking atual e o próximo
                     divida o intervalo em 4 partes iguais
                     para cada um dos eventos da trilha corrente que estão no intervalo
                     calculado faça
                            classifique o evento em uma das coleções: tempo,
                            primeiraSemi, colcheia, segundaSemi de acordo com o
                            intervalo de quatro partes iguais
                     crie um objeto com as coleções preenchidas e uma indicação dizendo
                     à que tempo aquele objeto pertence
       calcule as freqüências de ocorrência de eventos nas coleções
       calcule as médias de microdinâmica nas coleções
       calcule as médias dos desvios de microandamento nas coleções
```

Figura 1: Algoritmo de Microandamento e Microdinâmica

Com o objetivo de facilitar o entendimento do algoritmo da Figura 1, será demonstrado um exemplo prático do funcionamento do mesmo. Esse exemplo consiste no início da canção Barquinho tocado pelo intérprete 1 (dois primeiros tempos). A Figura 2 exemplifica o algoritmo.

$$\mathbf{003} \rightarrow 025 \leftarrow \underline{\mathit{047}} \rightarrow 069 \leftarrow \underline{\mathit{091}} \rightarrow 113 \leftarrow \underline{\mathit{135}} \rightarrow 157 \leftarrow \mathbf{180} \tag{1}$$

$$\mathbf{180} \rightarrow 201 \leftarrow \underline{\mathit{222}} \rightarrow 243 \leftarrow \underline{\mathit{264}} \rightarrow 285 \leftarrow \underline{\mathit{306}} \rightarrow 328 \leftarrow \mathbf{350} \tag{2}$$

Figura 2: Exemplo da segmentação de dois tempos de Barquinho

Em **negrito**, temos os *beat tracking* ou cabeças (em *ticks*) correspondentes à esses dois tempos (**003-180**: primeiro tempo; **180-350**: segundo tempo). Esses valores dos *beat tracking* foram obtidos por [de Lima, 2007] conforme explicado anteriormente. Já a parte em *itálico sublinhada*, corresponde aos valores da 2ª, 3ª e 4ª semicolcheias respectivamente e foram obtidos de acordo com a divisão binária proposta neste trabalho, ou seja, dividiu-se o tempo em 4 partes iguais (**180-003**)/4 que, truncado, corresponde a 44. Desta forma, temos a 2ª semicolcheia igual a 003+44=*047*, a 3ª semicolcheia igual a 047+44=*091* e a 4ª semicolcheia igual a 091+44=*135*. O mesmo procedimento é feito para o segundo tempo (**350-180**)/4, onde obtêm-se os valores: *222*, *264* e *306*. Todos esses valores correspondem ao exato momento em que as notas deveriam ser tocadas no caso de uma execução totalmente mecânica.

No entanto, para classificar os eventos como pertinentes à uma das coleções (cabeça, 2ª semicolcheia, 3ª semicolcheia ou 4ª semicolcheia) dividiu-se ao meio a distância entre duas coleções adjuntas e classificou-se os eventos como pertencentes à uma dada coleção de acordo com a localização do evento nessa divisão. Por exemplo, dividindo ao meio as seguintes coleções do primeiro tempo: 2ª semicolcheia (*047*) e 3ª semicolcheia (*091*) temos (*091*-*047*)/2=22, ou seja, os eventos menores que *047*+22=069 e maiores ou iguais a 025 (025 é o valor da divisão ao meio das coleções adjacentes cabeça e 2ª semicolcheia) serão classificados como pertencentes ao conjunto 2ª semicolcheia, os eventos maiores ou iguais a 069 e menores que 113 (113 é o valor da divisão ao meio das coleções adjacentes 3ª semicolcheia e 4ª semicolcheia) serão classificados na 3ª semicolcheia e assim sucessivamente. Toda essa classificação acima faz parte apenas do primeiro tempo, no entanto, essa mesma lógica passa a valer para os demais tempos.

Desta forma, ao final da primeira parte do algoritmo, teremos uma coleção de objetos que contém as seguintes informações: eventos ocorridos no tempo, eventos ocorridos na primeira semicolcheia, eventos ocorridos na segunda semicolcheia e eventos ocorridos na terceira semicolcheia. De posse desses dados, calcularemos as freqüências de ocorrência dos mesmos, bem como as médias de microdinâmica e as médias dos desvios de microandamento de acordo com alguns experimentos, como: calcular as médias tempo à tempo, calcular as médias compasso à compasso e, por fim, calcular as médias frase à frase.

4. Resultados

Foram analisados alguns clássicos da bossa nova (Tabela 1) tocadas por dois intérpretes, onde a única exigência foi tocar as canções seguindo cifras previamente fornecidas. A partir da análise das execuções, tentamos responder as seguintes questões: (a) há padrões de microdinâmica e microandamento? Em outras palavras, as variações de tempo e intensidade esboçam algum padrão? (b) ao estudar as variações de microdinâmica e microandamento deve-se utilizar qual janela de tempo: a unidade de tempo, o compasso ou a frase (que dura dois compassos)? Em outras palavras, tais variações sofrem influência da frase ou não?

É importante entender que a questão da janela de tempo da análise envolve uma questão mais sutil e importante: quando o músico executa (intuitivamente) as variações de microandamento e microdinâmica, ele é influenciado pela noção de frase, que representa um dos elementos-chave da estrutura musical?

Para isso, fez-se necessário um comparativo dos gráficos de análise por frase, por compasso e por tempo. Sabendo-se que, na bossa nova, a frase é formada por dois compassos e o compasso é formado por dois tempos, os gráficos mostrados a seguir correspondem a duração de uma frase (quatro tempos). Com o intuito de superpor as informações para melhor visualizar a análise, foi preciso duplicar as informações da análise por compasso (para corresponder a 4 tempos) e quadruplicar as informações da análise por tempo Os gráficos comparativos gerados, então, são divididos em quatro tempos e demonstram os valores (tomando a média total das notas tocadas no corpus para cada intérprete) dos microdesvios de tempo (Figuras 3 e 4) e de intensidade (Figuras 5 e 6). Cada tempo, como dito anteriormente, é dividido por quatro, sendo composto pela cabeça (T), 2ª semicolcheia (2ª semic.), 3ª semicolcheia (3ª semic.) e 4ª semicolcheia (4ª semic.).

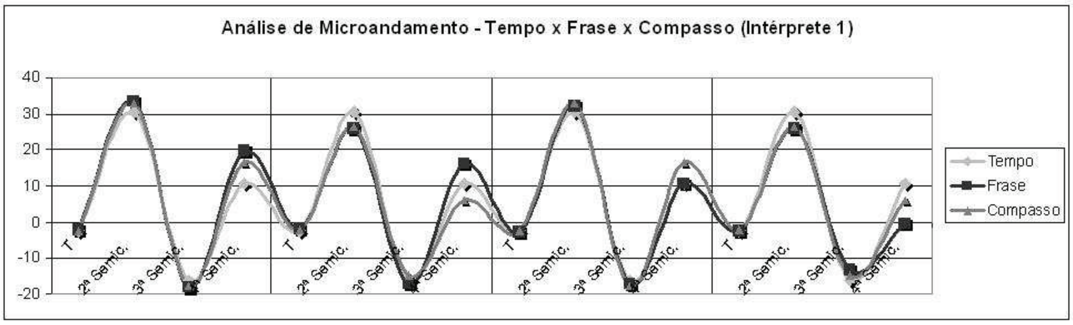


Figura 3: Análise de Microandamento: Tempo x Frase x Compasso (Intérprete 1)

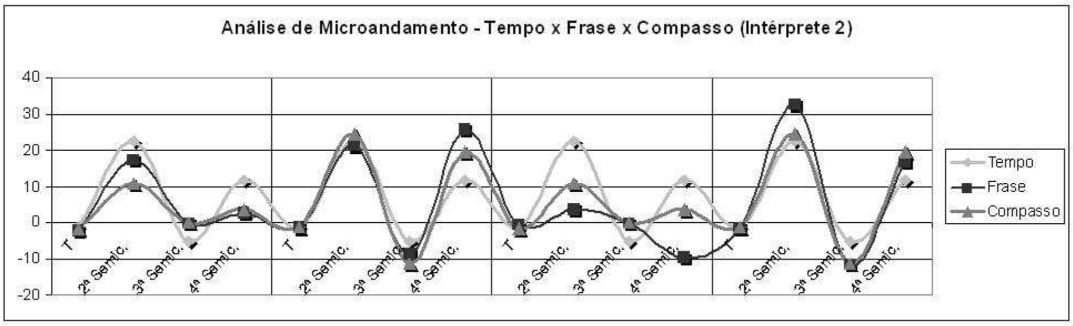


Figura 4: Análise de Microandamento: Tempo x Frase x Compasso (Intérprete 2)

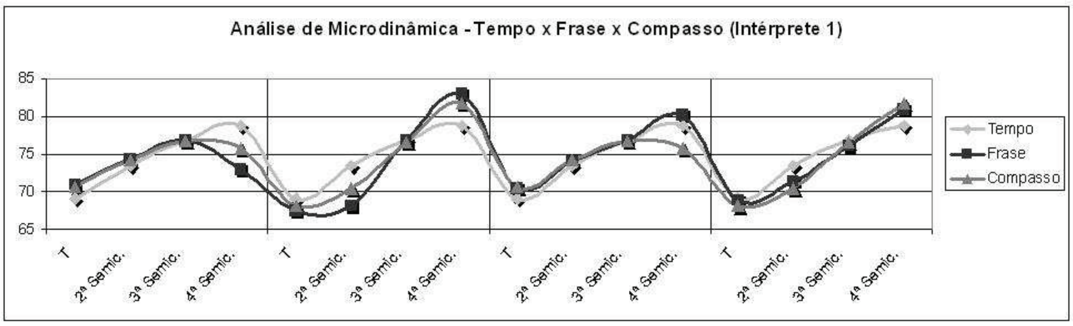


Figura 5: Análise de Microdinâmica: Tempo x Frase x Compasso (Intérprete 1)

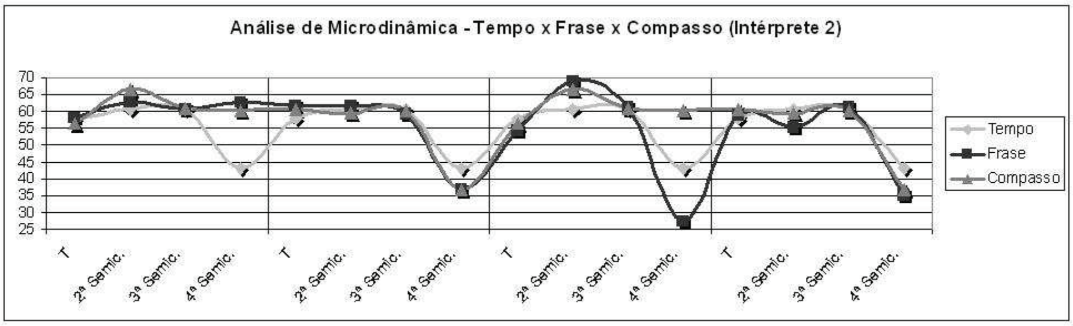


Figura 6: Análise de Microdinâmica: Tempo x Frase x Compasso (Intérprete 2)

Tendo em vista a primeira questão que buscávamos investigar, pode-se dizer que independentemente da janela de análise adotada e do intérprete em questão, emergem padrões de variação tanto do microandamento (na forma de um "M") quanto da microdinâmica (esses últimos mais complexos). Em outras palavras, as notas não são tocadas com igual microdesvio de tempo ou de intensidade. Dependendo de onde a nota ocorre, ela terá um microdesvio relativamente predefinido e diferente.

Quanto à questão da janela de tempo da análise, ou da influência da frase, deve-se estudar cada figura. Na Figura 3, observa-se que as diferenças (5ms a 10ms) entre janelas de análise acontecem sobretudo na última semicolcheia dos tempo. Pode-se erroneamente achar que essas diferenças são globalmentes desprezíveis, mas observando atentamente a forma da curva das medidas dentro da "frase", fica claro que essa forma muda do primeiro tempo para o segundo, do segundo para o terceiro e do terceiro para o último. Igualmente, comparando-se a forma da curva da frase no primeiro compasso (tempos 1 e 2) com a do segundo compasso (tempos 3 e 4), também uma diferença aparece. Portanto, mesmo se de forma relativamente discreta, a frase parece ter uma influência na forma de executar as microvariações de tempo e intensidade para o intérprete 1.

Já na Figura 4, correspondente ao segundo intérprete, as diferenças saltam aos olhos, dispensando uma análise mais minuciosa.

Considerando agora a análise de microdinâmica, embora representem formas diferentes, tanto no primeiro quanto no segundo intérprete, há clara diferenças entre as variações de tempo e intensidade segundo a janela de tempo escolhida.

5. Conclusão

Apresentamos um trabalho original sobre a abordagem violonística da bossa nova focado no estudo das pequenas variações de tempo e intensidade, respectivamente microandamento e microdinâmica. Essas duas variações são importantes, pois estão intimamente relacionadas à expressividade musical, que no caso do violão brasileiro é comumente associada a adjetivos como "molho", "levada", "groove", "swing".

Primeiramente, os experimentos mostraram que em ambos os intérpretes estudados, independentemente da janela de análise adotada, emergem padrões de variação do microandamento quanto da microdinâmica. Particularmente no microandamento, aparecem curvas em forma de "M" indicando um "sobe e desce" ou "vai e vem" na maneira de desviar-se no tempo, alternando entre "*ahead the beat*" e "*behind the beat*". É um padrão curioso, já que ao jazz normalmente a tendência é estar majoritariamente "*behind the beat*". Este resultado, apesar de precisar ser confirmado com um maior número de intérpretes, já representa um indício de conhecimento musicológico novo para o violão de bossa.

Quanto à questão da janela de análise, nos dois casos analisados, ficou clara a influência da noção de frase nas variações de microdinâmica. No caso do microandamento, a influência mostrou-se mais discreta no primeiro intérprete, porém bastante significativa no segundo. De forma geral, os experimentos reforçam a hipótese de que as variações de microandamento e microdinâmica dentro de uma frase não são as mesmas quando usada uma janela de análise com nível de granularidade menor (unidade de tempo e compasso). No entanto, uma conclusão definitiva sobre a questão vai depender da realização de captura e análise de outros intérpretes.

Se essa influência for confirmada, será uma grande novidade não só pelo que isso representa em termos de conhecimento etnomusicológico do violão de bossa, mas por representar um outro paradigma de análise. De fato, a literatura que estuda microandamento e microdinâmica costuma trabalhar com uma janela de análise de uma unidade de tempo, desconsiderando a dimensão estrutural da música, formada pelas frases, sessões, etc.

Estamos atualmente trabalhando na análise da influência dos (macro) padrões rítmicos no microandamento e microdinâmica. Os resultados preliminares mostram influência ainda mais forte do que no caso da frase. Confirmando-se tais resultados, haverá uma indicação de que aquilo que chamamos de padrão musical, formas recorrentes na música, tem várias dimensões, entre elas: um tempo macroscópico, um microandamento, uma microdinâmica, etc.

Referências

Binet, A. e Courtier, J. (1895). Recherches graphiques sur la musique. *LAnnee Psychologique*, 2:201–222.

Chediak, A. (1994). *Songbook Bossa Nova*, volume 5. Lumiar.

Clynes, M. (1995). Microstructural musical linguistics: Composer's pulses are liked by the best musicians. *Cognition*, 55:269–310.

de Lima, E. T. (2007). *Descoberta Automática de Conhecimento em Interpretações Musicais: O Caso do Acompanhamento Rítmico ao Violão*. Tese de doutorado, Universidade Federal de Pernambuco.

Dixon, S. (2001). Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58.

Ebhardt, K. (1898). Zwei beiträge zur psychologie des rhythmus und des tempos. *Zeitschrift für Psychologie und Physiologie des Sinnesorgane*, 18:99–154.

Freeman, P. e Lacey, L. (2002). Swing and groove: Contextual rhythmic nuance in live performance. Em *Proceedings of the 7th International Conference on Music Perception and Cognition (ICMPC7), Sydney*.

Friberg, A. e Sundström, J. (2002). Swing ratios and ensemble timing in jazz performances: Evidence for a common rhythmic pattern. *Music Perception*, 19(3):333–349.

Gabrielsson, A. (1999). Music performance. Em Deutsch, D., editor, *Psychology of Music*, pp. 501–602, San Diego. Academic Press.

Goebl, W., Pampalk, E., e Widmer, G. (2004). Exploring expressive performance trajectories: Six famous pianists play six chopin pieces. Em *Proceedings of the 8th International Conference on Music Perception and Cognition (ICMPC8), Evanston*, pp. 505–509. Adelaide.

Gouyon, F. (2007). Microtiming in "samba de roda- preliminary experiments with polyphonic audio. Em *Proceedings of the Brazilian Symposium on Computer Music*.

Júnior, R. C. S. N. (2006). Extração de informação simbólica de Áudio: Algoritmos de reconhecimento de onsets. Technical report. Trabalho de Conclusão de Curso.

Palmer, C. (1997). Music performance. *Annual Review of Psychology*, 48:115–138.

Scholz, R. (2008). Cochonut: Um processo para reconhecimento de acordes em seqüências capturadas por violões midi. Dissertação de mestrado, Universidade Federal de Pernambuco.

Sears, C. H. (1902). A contribution to the psychology of rythm. *American Journal of Psychology*, 13:28–61.

Sundberg, J., Friberg, A., e Frydén, L. (1991). Common secrets of musicians and listeners: An analysis-by-synthesis study of musical performance. Em Howell, P., West, R., e Cross, I., editores, *Representing Musical Structure*, pp. 161–197, London. Academic Press.

Sundberg, J. e Verrillo, V. (1980). On the anatomy of the ritard: A study of timing in music. *Journal of the Acoustical Society of America*, 68:772–779.

Todd, N. P. M. (1985). A model of expressive timing in tonal music. *Music Perception*, 3:33–58.

Todd, N. P. M. (1995). The kinematics of musical expression. *The Journal of the Acoustical Society of America*, 97(3):1940–1949.

Widmer, G. (2001). Using ai and machine learning to study expressive music performance: project survey and first report. *AI Commun.*, 14(3):149–162.

Widmer, G., Dixon, S., Goebl, W., Pampalk, E., e Tobudic, A. (2003). In search of the horowitz factor. *AI Mag.*, 24(3):111–130.

Widmer, G. e Goebl, W. (2004). Computational models of expressive music performance: The state of the art. *Journal of New Music Research*, 33(3):203–216.

Wright, M. e Berdahl, E. (2006). Towards machine learning of expressive microtiming in brazilian drumming. Em *Proceedings of the 2006 International Computer Music Conference*, pp. 572–575, New Orleans, LA.

Zanon, P. e Widmer, G. (2003). Learning to recognize famous pianists with machine learning techniques. Em *Proceedings of the 3rd Decennial Stockholm Music Acoustics Conference (SMAC'03)*, Stockholm, Sweden.

Pitch-class composition in the *pd* environment

Oscar Pablo Di Liscia[1], Pablo Cetta[2]

[1] Instituto Universitario Nacional del Arte, Area de Artes Multimediales, Buenos Aires, Argentina.

[2]Instituto Universitario Nacional del Arte, Area de Artes Multimediales, Buenos Aires, Argentina.

p.diliscia@iuna.edu.ar, p.cetta@iuna.edu.ar

Abstract. *This paper presents research on composition-assisted software using the Pitch-Class Sets technique [Babbitt, 1961] [Forte, 1974]. In order to use the mentioned techique on music composition and analysis, a Library of External Objects for the program Pure Data [Puckette, 2009] was developed. The Pitch-Class Set technique uses both the combinatorial and set theory to organize the twelve Pitch-Classes of the tempered system in Sets in order to exploit their structural properties on atonal music composition and analysis. A latter projection of this system explores the possibilities of disposition of the Pitch-Class Sets in the musical space producing Combinatorial Matrices [Morris, 1984, 1987].*

1. Presentation

The Pitch-Class Set technique uses both the combinatorial and set theory to organize the twelve Pitch-Classes of the tempered system in Sets (*Pitch-Class Set* will be from here on abbreviated as PCS) in order to exploit their structural properties on atonal music composition and analysis. Although it is clearly granted that this system was inspired on the European pre and post serial atonal music, it was initially developed by American composers and theorists [Babbitt, 1961] [Forte, 1974].

1.1 Pitch Class Sets.

Generally speaking, the PCSs technique covers three aspects: Taxonomy, Properties and Relations. The first aspect deals with the concept of the PCS as a subset of the Universal Superset formed by the twelve pitch-classes(also called “aggregate”) and the concepts of equivalence by inversion-transposition that generate the 224 different Set-Classes(*Set-Class* will be from here abbreviated as SC). The second defines, encodes, analyzes and classifies the structural features of each SC (as is, for example, its Interval Class Vector). The third deals with the possible relations between PCSs and SCs and their significance in the musical context (see, [Forte, 1974], [Morris, 1980], [Isaacson, 1990], [Rahn, 1980], among others).

1.2 Combinatorial Matrices.

A latter projection of this system explores the possibilities of disposition of PCSs in the musical space producing *Combinatorial Matrices* (*Combinatorial Matrix* will be from here on abbreviated as CM) and creating abstract compositional designs [Morris, 1984, 1987]. Such approach is based, of course, on the assumption that it is possible to extend some of the properties and relations of the PCSs to successions and combinations of them, a fact that is –in the opinion of the authors- proven by both the music and the theory of atonal music. In the theoretical works of Morris [Morris, 1984, 1987], the CMs are two-dimensional combinations of PCSs where the constancy of one or two SCs is regarded as a cohesive factor, as it may provide some sonic uniformity. The extension of this criterion to more than two PCSs is also possible, being the easiest way of accomplishing it the combination (by juxtaposition or superposition) of the "simple" CMs. This possibility will be shown and explained in the final section of this paper.

1.3 And what about music?

An outstanding number of significant relations of the PCSs theory with the organizations of other sound or music properties (such as register, duration, intensity and timbre, among others) have been already developed in serial music and its derivations (See, among other, [Stockhausen, 1959], [Babbitt, 1961, 1962], [Mead, 1987], [Roming, 2000]). These relations provide an important basis for the production of meaningful music, as they may allow the production of a truly multi-dimensional musical space.

2-Computer applications of the theory

The complexity of the atonal theory makes its practical application almost impossible without the aid of computer applications. As a matter of fact, the PCSs technique and its extensions seem to be developed in synergy with the software for musical composition and analysis. This research and development approach was followed in the project "*Desarrollo de aplicaciones informáticas para la organización de la altura temperada en la composición y análisis musical*" (Area Trans Departamental de Artes Multimediales, Instituto Universitario Nacional del Arte, 2007-2008). The results of the projects are several computer applications, publications, documentation and data that may be obtained at: http://www.iuna.edu.ar/departamentos/multimedia/investiga/sitio_analisis/

One of the computer applications developed in this project is the *pcslib* library to be used in the *pd* environment (*Pure Data*, [Puckette, 2009]). *pcslib* is a set of "external objects" that allows the work with PCSs and CMs in the *pd* environment. Generally speaking, the developed objects are software units that work with high level data types. The *pcslib* library is fully documented and some examples of the use of each object are provided as well. *pcslib* is an open source, and of free use, copy, and distribution. Among other factors, the *pd* environment was selected because it allows structured programming and because of its capability of handling Audio, MIDI and Graphic data in real and non-real time. The first feature ensures the possibility of working with different levels of data abstractions, which is especially useful in the applications of the PCSs and CMs theory to composition.

2.1 PCS and CM numerical representation in *pcslib*.

Because of both computing requirements and theoretical conventions, *pcslib* uses a numerical representation of the attributes of the PCSs and CMs which will be briefly described in this section. Since the code of *pcslib* is available, a user who is experienced in *c* programming language can find further details easily, if necessary.

A PCS is a set of *Pitch Classes*(PCs), each one of these being represented by one integer(from 0 to 11). Usually, a PCS is considered as *non-ordered*, meaning that 0,1,2,3 is equivalent to 0,2,1,3 and also to 0,1,3,2 and so on. However, in order to work properly with *Chains* and *CMs* it is necessary to consider the PCSs as *partially ordered*, in the sense that a PCS may have different groupings of its PCs referred to as *positions*. Such partially ordered PCS is referred to as a *chain*. A position of a chain may have more than one PC as well.

For example, the same PCS {0,2,4,5} may be distributed forming different *chains* with different number of positions(separated by spaces in what follows):

0245			(just one position with the four PCs)
02	45		(two positions with two PCs each)
024	5		(two positions with three and one PCs each respectively)
0	245		(two positions with one and three PCs each respectively)
0	24	5	(three positions with one, two, and one PCs each respectively)

In order to reflect numerically this ordering, *pcslib* represent an *end of position* with a -1 and each *end of chain* with a -2. Thus, the former examples of chains will be internally represented in *pcslib* as:

0245	-2				
02	-1	**45**	-2		
024	-1	**5**	-2		
0	-1	**245**	-2		
0	-1	**24**	-1	**5**	-2

A CM (Combinatorial Matrix) is a bi-dimensional array of PCs. X being a CM, the content of each of its positions may be accessed at $X_{row,column}$. The PCS content of each column and row of a CM must belong to the same SC or either to two different SCs[See Morris, 1984, 1987]. The simplest of the CMs is the one called *roman square*. An example of this CM is shown below:

0	1	3	4
1	3	4	0
3	4	0	1
4	0	1	3

Here we have a CM with 4 rows and 4 columns(4*4=16 positions) and only one PC in each position(which is, of course, not always the case).

A CM is represented by *pcslib* having as many vectors of 9 positions each as positions the CM have. Any position of each vector which is empty (i.e., with no PC in it) will be filled with a -4. For example, the former *roman square* CM shown above will be represented internally by *pcslib* as:

[**0** -4 -4 -4 -4 -4 -4 -4 -4] [**1** -4 -4 -4 -4 -4 -4 -4 -4] [**3** -4 -4 -4 -4 -4 -4 -4 -4] [**4** -4 -4 -4 -4 -4 -4 -4 -4]
[**1** -4 -4 -4 -4 -4 -4 -4 -4] [**3** -4 -4 -4 -4 -4 -4 -4 -4] [**4** -4 -4 -4 -4 -4 -4 -4 -4] [**0** -4 -4 -4 -4 -4 -4 -4 -4]
[**3** -4 -4 -4 -4 -4 -4 -4 -4] [**4** -4 -4 -4 -4 -4 -4 -4 -4] [**0** -4 -4 -4 -4 -4 -4 -4 -4] [**1** -4 -4 -4 -4 -4 -4 -4 -4]
[**4** -4 -4 -4 -4 -4 -4 -4 -4] [**0** -4 -4 -4 -4 -4 -4 -4 -4] [**1** -4 -4 -4 -4 -4 -4 -4 -4] [**3** -4 -4 -4 -4 -4 -4 -4 -4]
(PCs are marked boldface, and each [...] enclose the data representing a position of the CM)

A CM may have one or more positions with more than one PC in it, and/or one or more empty positions as well. For example, the 6X6 CM which follows combines positions of 3 and 2 PCs each. The CM shown below have several empty positions as well (positions [1][0], or [0][2], for example). An empty position is represented by *pcslib* as one -3 followed by eight -4 values. Note also that, in order to use only one character for each integer, an "A" replacing the integer 10 and a "B" replacing the integer 11 are used, but this is only for "printing neatness", not for the internal representation of the data.

01	**376**	**-**	**-**	**-**	**-**
-	**23**	**589**	**-**	**-**	**-**
-	**-**	**45**	**7AB**	**-**	**-**
-	**-**	**-**	**67**	**019**	**-**
-	**-**	**-**	**-**	**89**	**23B**
145	**-**	**-**	**-**	**-**	**AB**

2.2 Special data types used by *pcslib* under *pd*:

In order to perform an efficient data transfer between its different objects, in addition to the *pd* usual data types(like *floats*, *lists*, *symbols*, etc.), *pcslib* uses two special data types.

1-Pointers to structures of the PCS type.
2-Pointers to structures of the CM type.

Details of each type of structure can be found in the *pcslib* code. Of course, the data of each one of these structures cannot be accessed directly, but there are objects especially dedicated to create, read and modify them.

2.3 Summary of *pcslib* externals:

Due to the extension limitations of this paper, a detailed description of each of the 17 externals objects of *pcslib* will not be done. Therefore, this section will only present a summary of them. Two brief examples of use are provided in the next section.

Pitch-class sets(PCS) objects:

pcs_write: The *pcs_write* object generates a PCS using data delivered by the user. The PCS is scanned at the PCS table and transposed and/or inverted, if required.

pcs_read: The *pcs_read* object reads and classifies the data in a PCS structure.

pcs_pf: The *pcs_pf* object finds the prime form of a PCS and stores all its associated data in a PCS structure.

pcs_ttos: The *pcs_ttos* object performs the Transposition, Inversion followed by transposition or Multiplication operations on a PCS.

pcs_subs: The *pcs_subs* object gets the all the subsets of cardinal *n* out of a given PCS.

pcs_sim: The *pcs_sim* object evaluates the similarity degree of two SCs according to either Forte, Morris or Isaacson´s criteria(See [Forte, 1974], [Morris, 1980], [Isaacson, 1990]). In addition it also checks if the two PCSs belong to the same Kh complex(See [Forte, 1978]).

pcs_invar: The *pcs_invar* object finds the shared PCs between two or more PCSs.

pcs_parts: The *pcs_parts* object gets all the binary partitions out of a given PCS.

Combinatorial Matrix(CM) objects

cm_roman: The *cm_roman* object creates a *roman square* type of CM(see [Morris, 1984, 1987]).

cm_t1a: The *cm_t1a* object creates a *t1a* type of CM(see [Morris, 1984, 1987]).

cm_t1b: The *cm_t1b* object creates a *t1b* type of CM(see [Morris, 1984, 1987]).

cm_t2: The *cm_t2* object creates a *t2* type of CM(see [Morris, 1984, 1987]).

cm_opcy: The *cm_opcy* object creates a CM using cycles of operators(see [Morris, 1984, 1987]).

cm_2txt: The *cm_2txt* object has no output, only post to the *pd* prompt a "clean" i.e., not raw, version of a CM.

cm_read: The *cm_read* object gets the "raw" data in a CM. -3 and -4 are spaces and empty positions, respectively.

cm_2pcs: The *cm_2pcs* object gets a PCS from a CM.

cm_trans: The *cm_trans* object performs several transformations on a given CM. The transformations that may be performed by this object are: transposing, inverting, rotating by the diagonal, rotating by 90 degrees, exchange of rows or columns content and swapping elements to decrease density.

2.4 Two examples

In order to show some of the *pcslib* potential, two examples are presented in this section. Although the output of the *pd* patches shown is numerical and highly abstract, the readers must keep in mind that the goal of *pcslib* is to provide a toolkit for generating a control-signal flow which may be useful to drive a digital musical system. The design of such musical systems, however, implies large patches using the MIDI and Audio capabilities of *pd* and is not the subject of this paper.

First, a simple *pd* patch using the *pcs_pf* and *pcs_read* externals is shown on Figure 1. This simple patch intends to show just one way in which the PCS data type may be created, transmitted and read in order to access its data members, as well as introduces the reader in some of the basic concepts of the PCSs theory. In the patch, a list containing Pitch-Classes is passed to the *pcs_pf* object, which creates a PCS whose pointer is passed to the *pcs_read* object. The latter object outputs the following data by its seven outlets (ordered from left to right):

outlet1: The original PCs of the PCS as delivered (*floats* list).
outlet2: The cardinal number (*float*), i.e., the number of different PCs that the PCS have.
outlet3: The ordinal number (*float*) of the PCS. A number indicating the SC to which the PCS belongs. The Atonal Theory uses the cardinal and the ordinal numbers separated by a hyphen to define the name of the PCS and its SC (See [Forte, 1974]). In the example shown, the PCS name is "5-17".
outlet4: The status (T/I) (*symbol*). The transposition and/or inversion status of the PCS.
outlet5: The prime form of the PCS (*floats* list). The *prime form* of a PCS is a version of it determined by convention (See [Forte, 1974]).
outlet6: The interval-class vector (*floats* list). The six floats delivered represent the interval contents (interval-class content) of the PCS. The data is to be interpreted as a six-position numerical vector. Thus, the first number is the amount of members of the interval-class 1 (minor second, or SC 2-1), the second one is the amount of members of the interval-class 2 (major second, or SC 2-2), and so on.
outlet7: The literal complement of the PCS, i.e., the PCs that are not included in it (*floats* list).

It is important to note that, once a PCS structure is created, all its data may be useful to perform more complex operations in a signal flow which may drive a musical system. Just as an example, a patch could be done in which a minimal number of members of any interval-class of a PCS is required to take an action. Another possibility could be to compare two or more PCSs (using the *pcs_sim* object) in order to check whether a certain kind of relation stands.

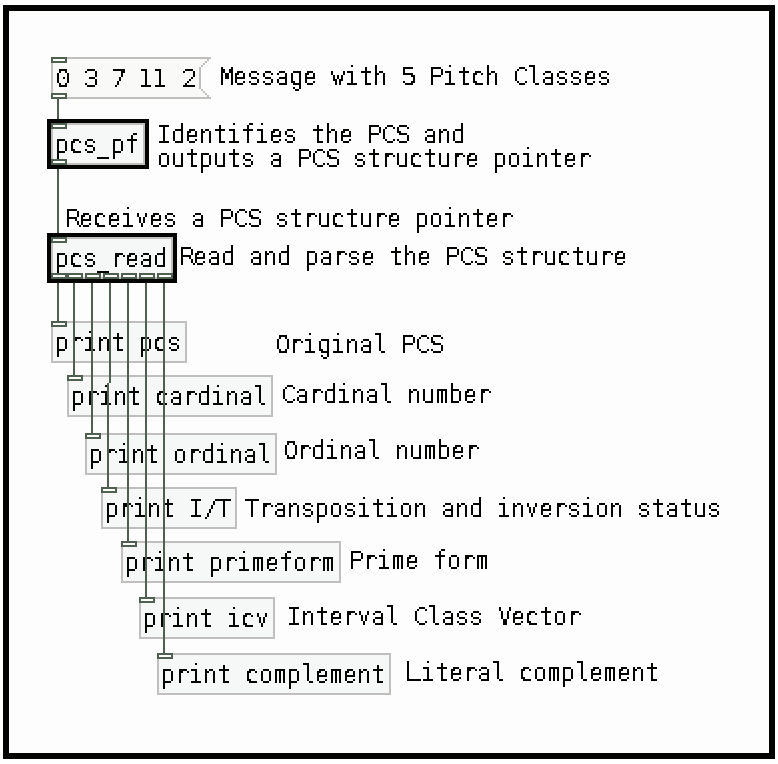


Figure 1: A *pd* patch using the *pcs_pf* and *pcs_read* objects.

The second patch presented uses the combination of PCSs in order to generate a CM, and performs several transformations of it. Here, a list containing Pitch Classes is passed to the *pcs_pf* object, which creates a PCS whose pointer is passed to the *cm_opcy* object. The latter object creates a CM and passes its pointer to the *cm_trans* object which will perform the transformations that are requested by the messages that are allowed to process. The *cm_2txt* shows the resulting CM and its successive transformations on the *pd* prompt. The patch is shown in Figure 2.

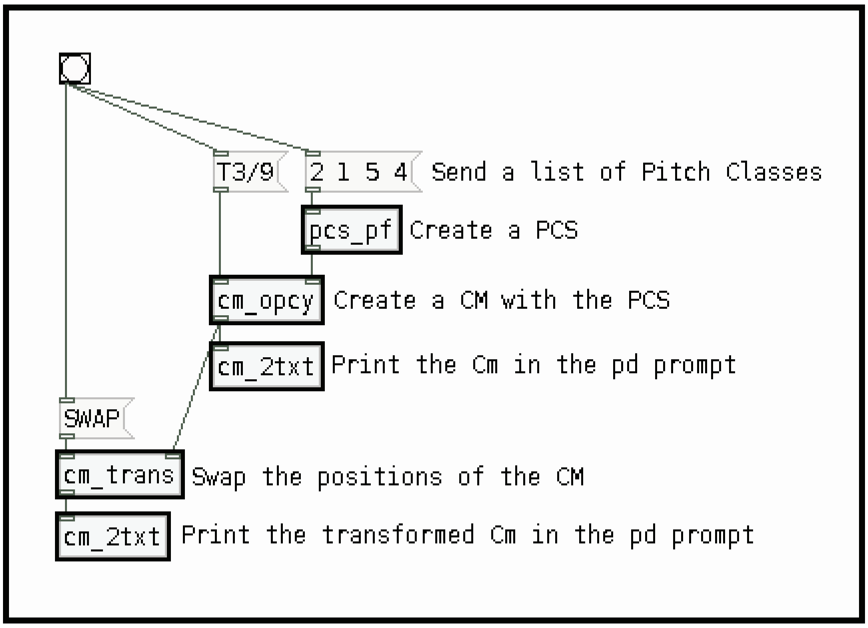


Figure 2: A *pd* patch using the *cm_opcy* and cm_*trans* objects.

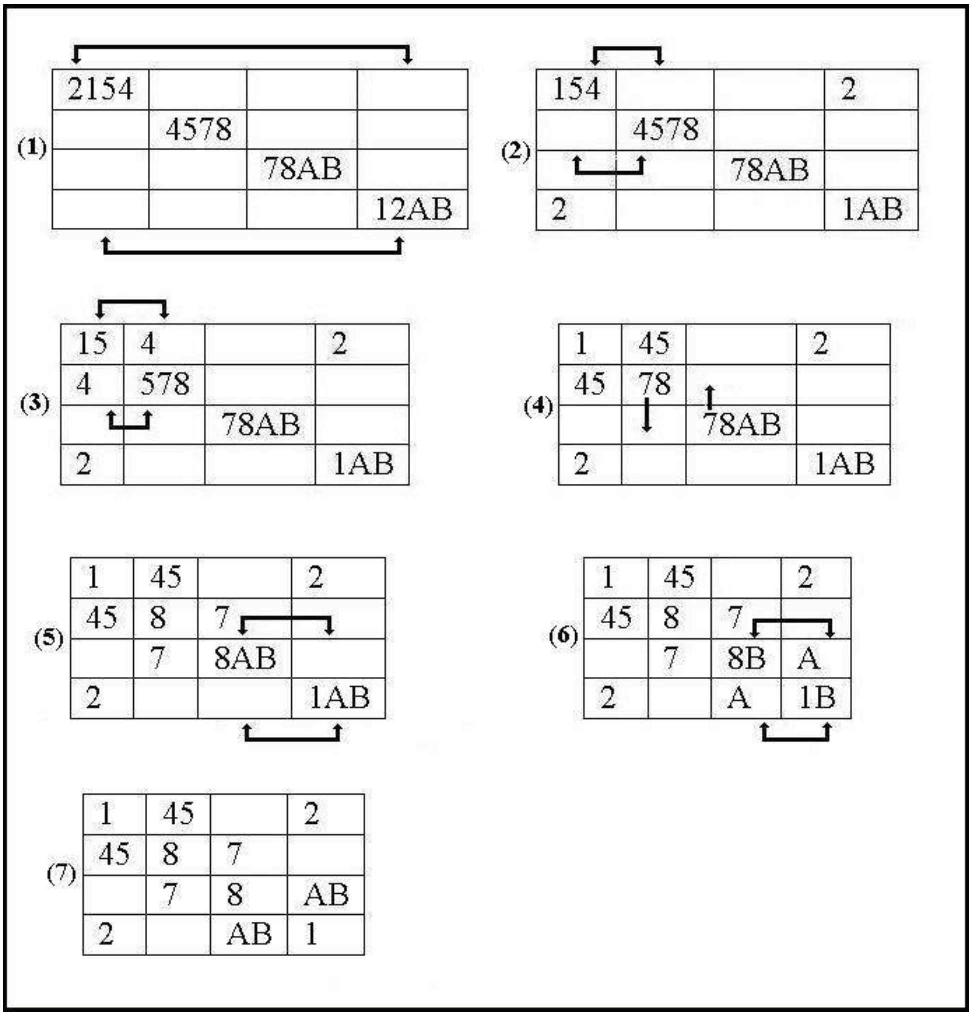


Figure 3: Plot of successive swapping operations on a CM.

In this case, a series of swapping operations was selected to get different variations of the original CM. The CMs resulting from the successive swapping operations are shown in Figure 3. It can be easily seen that the swapping operation is produced by the object *cm_trans* without changing the PCS content of the CM, thus producing a decrease in density(the amount of PCs in each position of the CM) while preserving the sonic uniformity.

The latter structures can be presented in different ways in order to generate a meaningful music sequence. Just to show one of them, the same time-interval is assigned to each PC in the CM, each one of its rows is assigned to a pitch register, and the beginning of each CM row is marked with an accent. The sequence shows an increasing density in simultaneity of notes and accents. Also, it can be seen that the time pattern of the accents is homogeneous at the beginning and the end of the sequence whilst it is non-homogeneous at its middle. The musical sequence is scored in two staffs only to keep it easier for reading and is shown in Figure 4.

Figure 4: A musical sequence using the output of the patch shown in Figure 2.

3-Conclusions

Though *pcslib* is at present still under development, an important number of external objects that are capable to perform operations of significant complexity have already been developed. The CM generation algorithms are effective and capable of generating a great number of them, but many of these are trivial and/or tautological. This does not mean that they are not useful, as it was shown by the examples that they constitute a strong base for creating pitch-organised music. It was also shown that, in order of having music and not merely PCSs and/or CMs, still higher level (i.e., more "close to music") programming is needed. Some of the directions to be further explored in the latter approach may include the development and implementation of criteria for the selection and combination of CMs.

4-References

Babbit, Milton (1961) “Set Structure as Composicional Determinant”, Journal of Music Theory 5, no.1, USA.

Babbit, Milton (1962) “Twelve-Tone Rhythmic Structure and the Electronic Medium”, Perspectives of New Music, no.1, USA, pp. 49-79.

Forte, Allen (1974) “The Structure of Atonal Music”, Yale University Press, England.

Isaacson, Eric (1990) “Similarity of Interval-Class Content between Pitch-Class Sets: The IcVSIM Relation”, Journal of Music Theory, Vol. 34 N° 1, USA, pags. 1/28.

Mead, Andrew (1987) About Time's Time: A Survey of Milton Babbitt's Recent Rhythmic Practice, Perspectives of New Music, Vol. 25 N°s 1/2, USA, pags. 182-235.

Morris, Robert (1980) “A similarity Index for Pitch-Class Sets”, Perspectives of New Music, Vol. 18 N°s 1/2, USA, pags. 445/460

Morris, Robert (1984) “Combinatorialty without the aggregate”, Perspectives of new Music, USA.

Morris, Robert (1987) “Composition with Pitch-Classes: A Theory of Compositional Design”, Yale University Press, USA.

Puckette, Miller(2009), “PD Documentation”, *http://crca.ucsd.edu/~msp/Pd_documentation/*

Rahn, John (1980) “Basic Atonal Theory”, Schirmer Books, USA.

Roming, James, (2000) “Twelve-tone rhythmic structure and its application to form: time-point nesting and rotation in spin”, Doctoral Thesis, Graduate School—New Brunswick Rutgers, The State University of New Jersey, USA.

Karlheinz Stockhausen, (1959) "...how time passes...", Die Reihe, Vol.III, Theodore Presser co., Pensylvania, 1959, USA.

Derivation of SOM-G Granular Synthesis Instruments from Audio Signals by Atomic Decomposition

Paulo R. G. da Silva

pgonsilva@gmail.com

Abstract. *The derivation of SOM-G granular synthesis instruments from recorded sounds by an analysis system based on the matching pursuit algorithm is presented. The implementation of the matching-pursuit algorithm and the structure of the dictionary of Gabor atoms are discussed. Audio signals recorded from acoustical musical instruments are analysed and compared with the reconstructed signals.*

1. An Analysis and Synthesis Experiment in Granular Synthesis

This paper presents the implementation of a system of analysis and synthesis of audio signals based on an atomic model of signal. The decomposition of audio signals into Gabor atoms is done by an implementation of the matching-pursuit algorithm. The atoms that result from the decomposition of an input signal are coded as an instrument in SOM-G language, that can be rendered into an audio signal and allowing for comparison between original and reconstructed signals.

The main objective of the implementation of the matching pursuit algorithm described in this paper is to obtain decompositions of a signal over a dictionary of Gabor atoms which duration is less than 100ms. Such durations are appropriate to the SOM-G instruments definition as will be discussed. The modelling of the sound of acoustical musical instruments as SOM-G instruments was the motivation for this implementation. So, obtain a compact decomposition of a signal is desirable, but was not the main objective of the implementation. The implementation can handle signals of different time-frequency characteristics. This is required for the decomposition of recorded samples from acoustical instruments that usually have transients and almost stationary parts in the same signal.

2. Introduction

The physicist Denis Gabor stated that a signal could be represented by a linear combination of elementary signals, named atoms or acoustical quanta [Gabor 1946]. He proposed a signal model in which time-domain and frequency-domain information are not dissociated, and suggested that the expansion in terms of atoms was more meaningful than Fourier analysis because the signals was considered simultaneously in time and frequency domains [Gabor 1947].

The model of Gabor inspired the synthesis technique named Granular synthesis, in which a signal is composed by a large number of short duration sounds named grains or atoms [Roads 1988]. Xenakis was the first to explain a compositional theory for granular synthesis [Xenakis 1963]. He proposes a possible approach to the model of Gabor in the context of an analog synthesis implementation, using sinusoidal waves of around 40 ms of duration modulated by rectangular envelopes. Curtis Roads systematically researched granular synthesis between 1975 and 1981, and is responsible for the first effective implementation of the technique [Roads 1987], [Roads 1988]. Barry Truax made the first real time granular synthesis experiment using a digital signal processing hardware [Truax 1988]. The difficulties on the generation and regulation of grains in granular synthesis has been evidenced since the first implementations, as it is usually necessary hundreds or thousands of grains per second to produce granular events. The active research on granular synthesis in the last years brought up various approaches to grains generation and regulation, and granular synthesis was used to create entirely new sounds. Several new approaches were developed. Some few examples show the variety of new approaches to granular synthesis regulation: cellular automata as granular regulation mechanism [Miranda 1995], granulation and synthesis from natural sounds as granular generation, allowing time or pitch transformations [Jones and Parks 1988],[Truax 1994], [Keller and Truax 1998], applications of group theory to granular synthesis [Fabbri and Maia Jr 2007], among other works.

Analysis-synthesis systems provide a conceptual framework for the development of signal modelling methods and their applications. The existence of a feasible analysis method for granular synthesis allows that the analysed signal be compared with the reconstructed signal so that the atomic model and the implementation can be tested.

There are some analysis methods that can derive time-frequency signal models. The Wavelet transform can be used to extract time-frequency information from audio signals [Kronland-Martinet 1988],[Faria 1997]. Basis pursuit applies modern linear algebra techniques to decompose a signal into an optimal combination of atoms chosen from a base [Chen, Donoho, and Saunders 1998]. Matching-pursuit [Mallat and Zhang 1993] is a greedy algorithm for the atomic decomposition in terms of atoms chosen from a dictionary.

The matching-pursuit algorithm is the analysis method that was implemented in the system described in this paper because its simplicity, stability and flexibility. Some improvements on the performance of the original algorithm has been presented, like Fast Matching Pursuit [Gribonval 2001] and Harmonic Matching Pursuit [Gribonval, Bacry 2003]. Improvements on the resolution of the analysis were brought by High Resolution Matching Pursuit [Gribonval, Bacry, Mallat, Depalle, Rodet 1996], and a measure of the destructive interference between atoms can be found in [Shynk, Daudet and Roads 2008].

3. Gabor Atoms

The greatest part of the theory of communication of the early twentieth century was developed on the basis of Fourier theorem. According to Gabor, though the Fourier method is mathematically correct, the physical interpretation of the results is somewhat difficult to reconcile with physical intuitions [Gabor 1946]. For human hearing, time and frequency patterns are associated in sound perception, but in Fourier theory time and frequency domains are mutually exclusive.

Gabor proposed a signal representation that reveals both its time and frequency structures. All the mathematical development can be found in [Gabor 1946] and [Gabor 1947], and we will just highlight the main results. The time frequency localization of each atom is constrained by a resolution limitation similar to the Heisenberg uncertainty principle of quantum mechanics.

$$\Delta t \Delta f \geq 1 \quad (1)$$

The inequality in (1) establishes an important relation between time and frequency resolution. In order to achieve the best time and frequency discrimination, the ideal form of the elementary signals should be one for which the product $\Delta t \Delta f$ has its minimal value and the inequality (1) becomes an equality. The signal for which $\Delta t \Delta f$ is unitary is the product of a harmonic oscillation by a Gaussian pulse.

$$\psi(t)=e^{-\alpha(t-t_0)^2}e^{i(2\pi*f_0*(t-t_0))} \quad (2)$$

The parameter fo is the mean frequency of the atom, and to is the mean epoch. The parameter α is related to the dilation of the pulse that modulates the harmonic oscillation, and determines the effective duration of the atom and its effective frequency bandwidth.

$$\Delta t=\frac{\sqrt{\pi}}{\alpha} \quad (3)$$

$$\Delta f=\frac{\alpha}{\sqrt{\pi}} \quad (4)$$

Real atoms must have an additional parameter, the phase shift φ of the harmonic oscillation. The mathematical form of real Gabor atoms is shown by expression (5).

$$\psi_r(t)=e^{-\alpha(t-t_0)^2}\cos(2\pi*f_0(t-t_0)+\varphi) \quad (5)$$

Figure 1 shows the aspect of a real Gabor atom for α = 20, fo=110 and φ=0. This value of α implies in Δt=88,2 milliseconds and Δf=11,28 Hertz. The dotted line represents the gaussian function that modulates the harmonic oscilation.

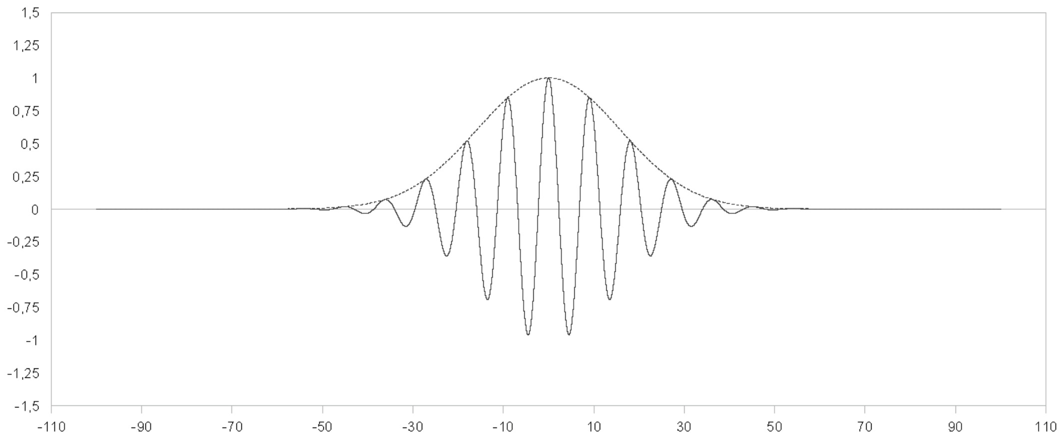


Figure 1 – A real Gabor atom.

Each atom can be represented as a rectangle in a time x frequency diagram . The center of the rectangle stays at the coordinates of the mean epoch and mean frequency; its width is proportional to its effective duration Δt and its height is proportional to its effective bandwidth Δf. Such diagram is called an information diagram, and the rectangles that represent atoms in an information diagram are called characteristic cells.

Figure 2 shows an information diagram and the representation of atoms as characteristic cells. The information diagram contains information about both time and frequency structures of a signal.

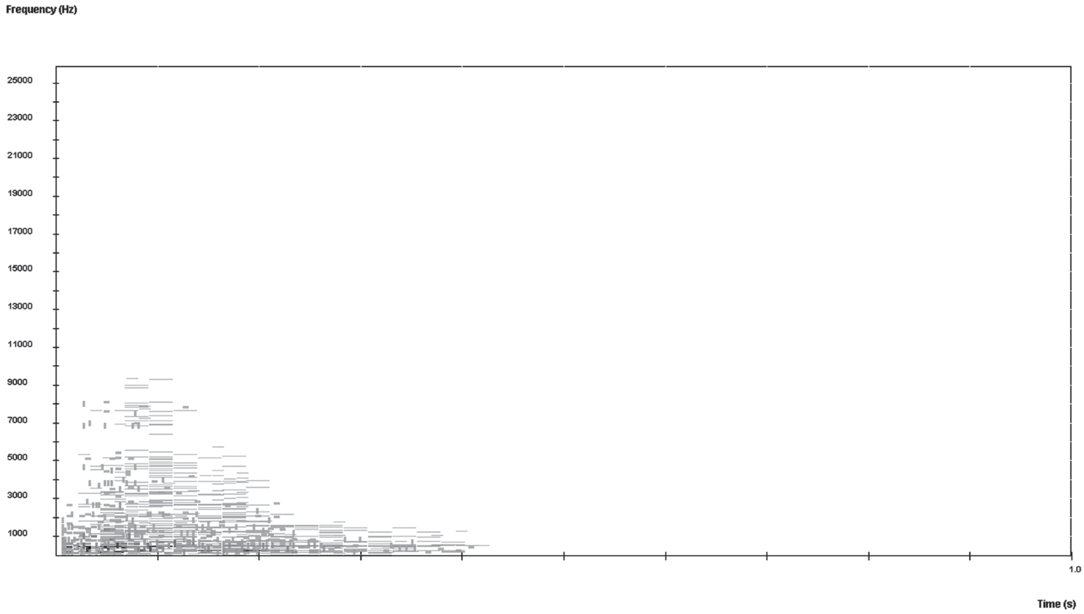


Figure 2 – The Information Diagram

4. Overview of the Matching Pursuit Algorithm

Matching Pursuit [Mallat and Zhang 1993] is a greedy iterative algorithm for deriving signal decompositions in terms of expansion functions chosen from a dictionary of basis functions or *atoms*. At each iteration, the algorithm looks in the dictionary for the atom that best approximates the signal, where the two-norm is used as the approximation metric. The contribution of the chosen atom is then subtracted from the signal and the algorithm restarts to one more iteration over the residual, until some halting criterion is met, as a residual energy threshold. The mathematical development of the algorithm and the proof of its convergence can be found in [Mallat and Zhang 1993], and a comparison with other atomic decomposition methods can be found in [Goodwin 1997].

Let D be a dictionary of complex atoms. Each function $d_k \in D$ can be characterized by its effective duration δ, its mean epoch τ and its mean frequency f. Let all atoms in D be normalized

$$\langle d_k , d_k \rangle = 1 , \forall d_k \in D \quad (7)$$

The task at the i-th iteration of the algorithm is to find the atom $d_k \in D$ that minimizes the two-norm of the residual signal r_i. It can be shown that this is equivalent to choosing the atom whose inner product with the signal has the largest magnitude

$$d_i = arg\ max_{d_i \in D} |\langle d_i , r_i \rangle| \quad (8)$$

The i-th expansion coefficient α_i is the inner product between the chosen atom d_i and the residual signal r_i.

$$\alpha_i = \langle d_i , r_i \rangle \quad (9)$$

At the end of the iteration, the term $\alpha_i d_i$ is subtracted from the residual r_i

$$r_{i+1} = r_i - \alpha_i d_i \quad (10)$$

After I iterations, the signal S can be represented by the expression

$$S=\sum_{i=1}^{I} \alpha_i d_i + r_{I+1} \quad (11)$$

The mean-squared error of the reconstructed signal decreases as the number of iterations increase, so matching pursuit can derive a reasonable approximation for a signal. It is well-known that matching-pursuit does not lead to optimal approximations, but greedy approaches are justified given the complexity of finding an optimal approximation, a NP-Hard problem [Goodwin 1997].

With a dictionary of Gabor atoms, a matching pursuit defines a time-frequency transform. An appropriate dicionary is required to achieve compactness, but there is a compromise between the number of atoms present in a dictionary and the number of computations necessary to choose the atom that best fits the signal at each iteraction.

5. An Implementation of the Matching-Pursuit Algorithm

The matching-pursuit algorithm was implemented as a java package and integrated to the implementation of the SOM-G language packages. The result of the decomposition of an audio file is expressed as a SOM-G instrument. A granular analysis/synthesis system was implemented; the SOM-G interpreter can reconstruct the signal from the granular synthesis instrument obtained . Figure 3 shows a fluxogram for the decomposition of a signal.

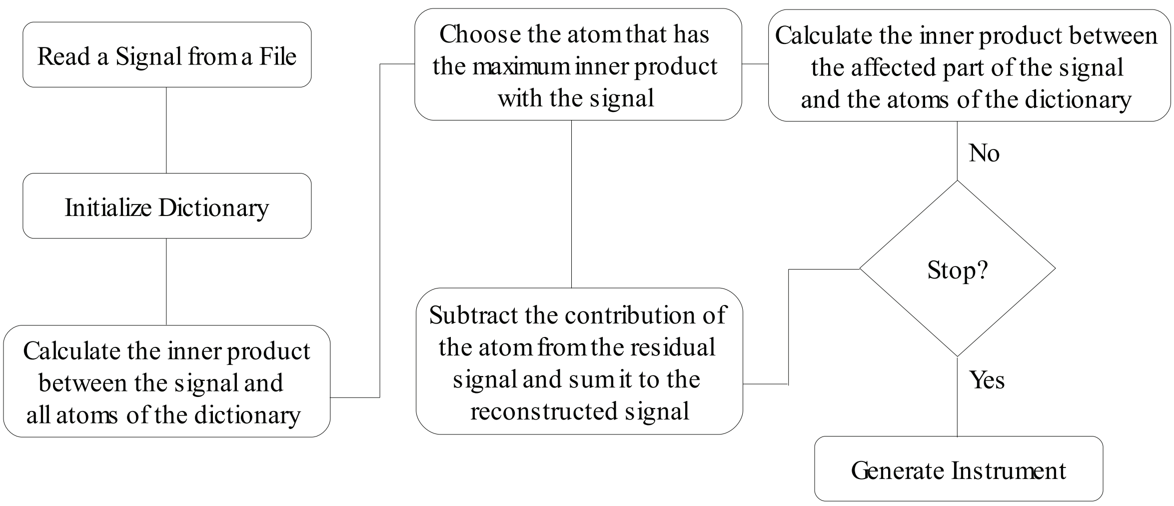


Figure 3 – Fluxogram of the decomposition process

In order to obtain an analytic signal from a real signal, a Hilbert transform is done over it. This is not a requirement of the matching-pursuit algorithm that can be implemented with real atoms by the introduction of a phase parameter in the dictionary, but complex atoms incorporates the phase as an implicit parameter and lead to a simpler algorithm. After the decomposition, the atoms can be converted to real signals and the phase can be extracted from the complex coefficients that result from the decomposition.

The computation of the correlations $\langle d_i, r_i \rangle$ for all $d_k \in D$ is costly, so the implementation previewed a strategy to avoid unnecessary computations. The atoms used in the implementation are finite, and each atom extracted from the residual signal affects only part of the signal. At each iteration the correlations are stored, and when the atom that has the largest magnitude of correlation is chosen, only the correlations in the part affected by the subtraction of its contribution must be recalculated for the next iteration. The class diagram of the package atomic_decomposition is shown in figure 4.

The dictionary composed only by Gabor atoms was constructed with only five effective durations for most of the signals that were decomposed: 3, 6, 12, 24 and 48 milliseconds. For each duration, the frequencies are distributed according to the interval calculated by the relation (1), from a minimal fixed value to half of the sampling rate of the analysed signal, according to Nyquist sampling theorem. The translation of the atoms are fixed as the effective duration of the atoms.

The diagram of the classes in the package atomic_decomposition is shown in figure 5. The package has only three classes: AtomicDecomposer, GaborDictionary and Signal.

The class *AtomicDecomposer* implements the matching pursuit algorithm. It has a constructor that accepts as argument a reference for an audio file. The code bellow shows the creation of an instance of the AtomicDecomposer class:

```
mp = new AtomicDecomposer(new File("sample.wav"));

mp.start();
```

The class *GaborDictionary* has its structure defined by an array that stores the durations in milliseconds of the grains:

```
durations[0]  = 0.003f;
durations[1]  = 0.006f;
durations[2]  = 0.012f;
durations[3]  = 0.024f;
durations[4]  = 0.048f;
```

A new instance of the GaborDictionary class can be created as follows.

```
/* Creates a Gabor Dictionary with minimum frequency of 15
Hz, maximum frequency of 44100 Hz and sample rate of 44100
Hz */
DC = new GaborDictionary(15, 22050, 44100);
```

The class *Signal* can represent a complex signal of one or two channels, and has many convenience methods. An example of the creation and initialization of an instance of the Signal class is shown bellow:

```
sg = new Signal(new File("sample.wav"));

sg.create_analytic_signal();
```

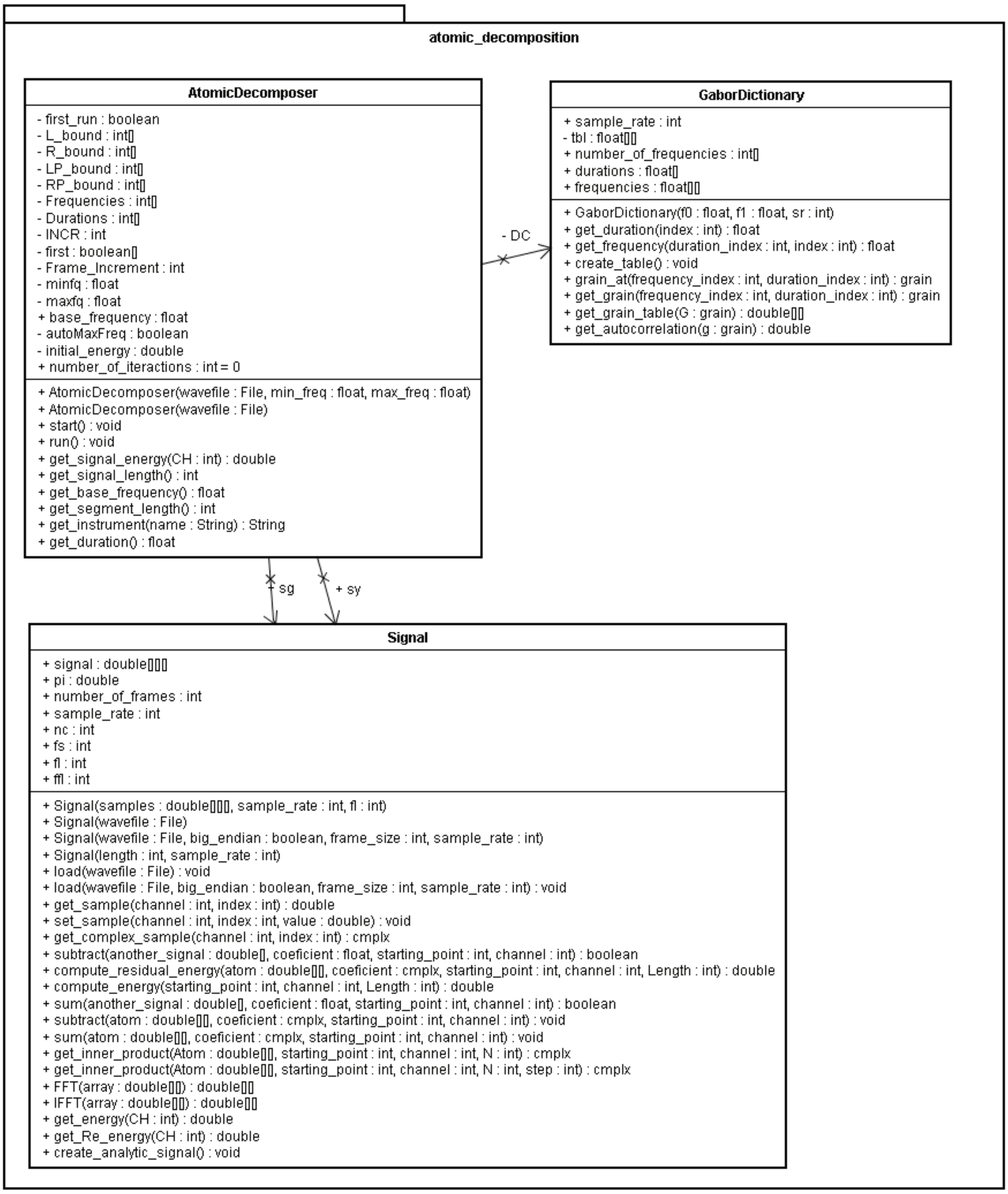


Figure 4 – Package atomic_decomposition – Class Diagram

6. Results

The decomposition and resynthesis of a berimbau note is shown bellow. A berimbau is an african percussion instrument. It has only one string, that is played with a wood stick and a rock.

Figure 5 shows the recorded signal. Figure 6 shows the reconstructed signal. Figure 7 shows the spectrum of the analysed signal, and figure 8 shows the spectrum of the resynthesized signal. The signal was recorded at 44100 Hz, 16 bits. The analysis resulted in 6965 grains for each channel.

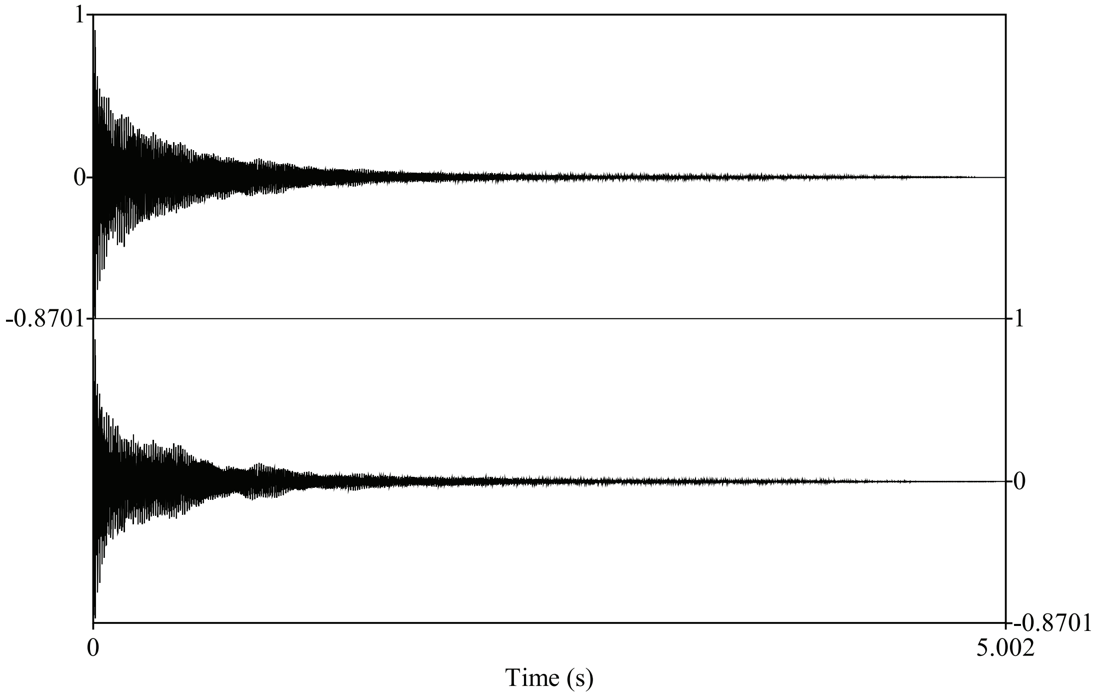


Figure 5 – The original recorded signal

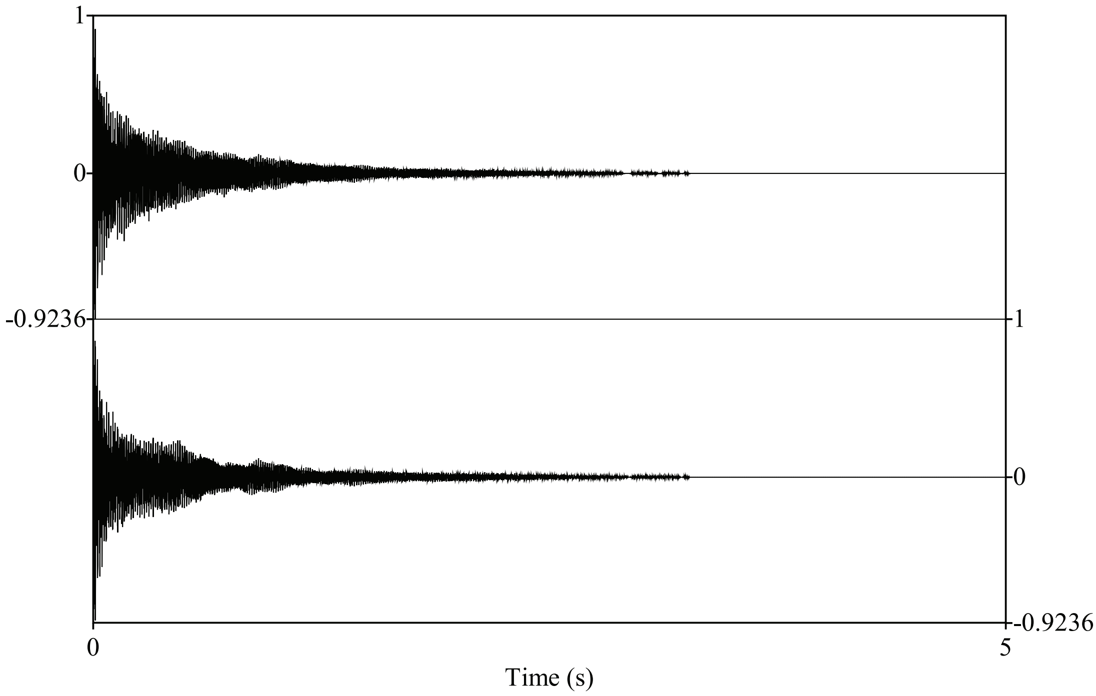


Figure 6 – The resynthesized signal

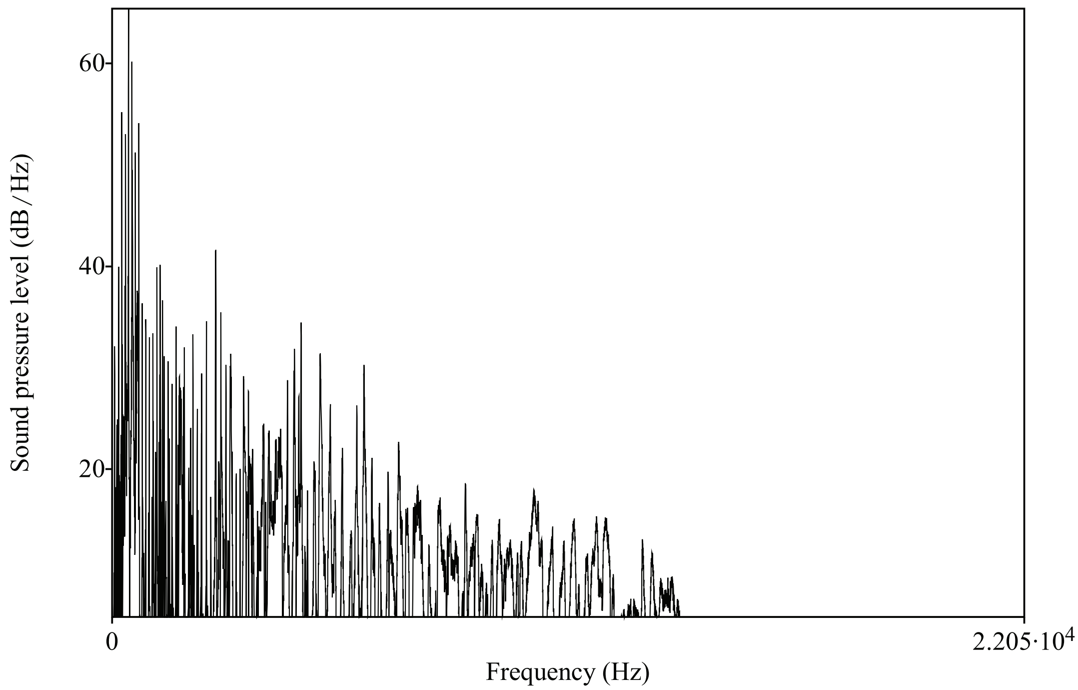


Figure 7 – The spectrum of the recorded signal

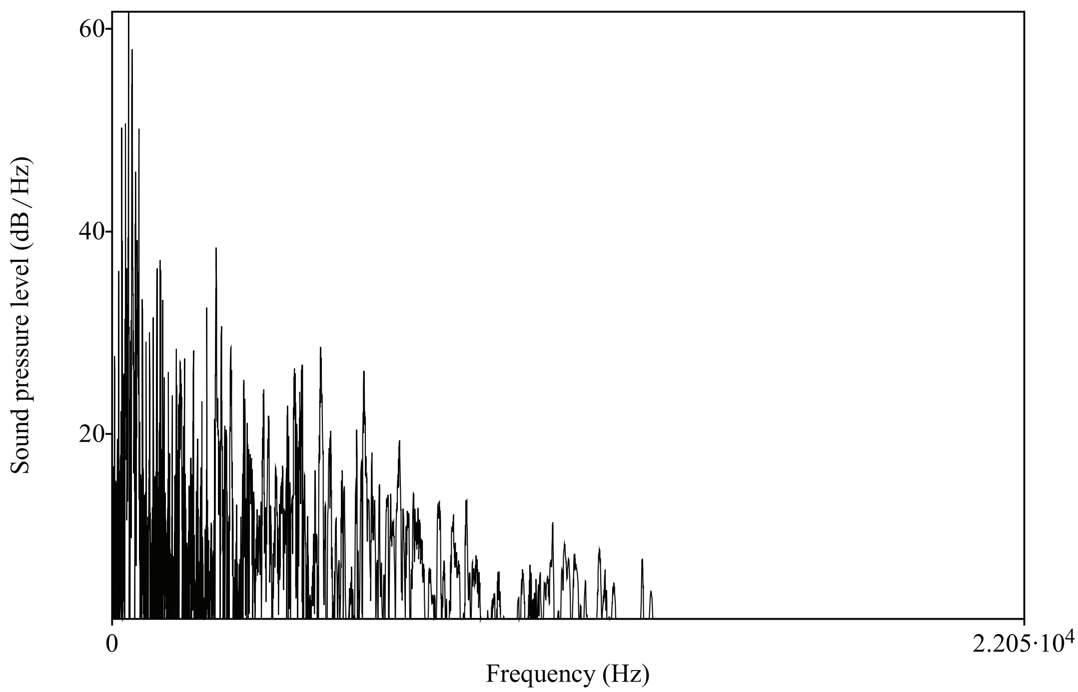


Figure 8 – The spectrum of the resynthesized signal

7. Future Work

A bank of granular synthesis instruments derived from acoustical instruments can be constructed and employed for music composition applications, improving the musical possibilities of the SOM-G language. A bank of phonems can also be modeled as granular synthesis instruments and applied to the design of speech synthesis systems.
An implementation of the whole system in a faster, non-interpreted language is desirable.

8. References

Chen, S. Donoho, D. and Saunders, M. "Atomic Decomposition by Basis Pursuit," SIAM, vol. 20, no. 1, pp. 33–61, 1998.

Fabbri, R. and Maia Jr, A. "Applications of Group Theory on Granular Synthesis". Annals of the VIII Brazilian Symposium on Computer Music, 109-120, 2007.

Faria, R. R. A. "Aplicação de Wavelets na Análise de Gestos Musicais em Timbres de Instrumentos Acústicos Tradicionais" Msc thesis, Universidade de São Paulo,1997.

Gabor, D. "Theory of Communication". J. Inst. Elec. Eng. (London) 93,429-457, 1946.

Gabor, D. "Acoustical Quanta and the Theory of Hearing". Nature 4044,591-594,1947.

Gonçalves, P. and Arcela, A. "SOM-G, a Language for Granular Synthesis". Annals of the VIII Brazilian Symposium on Computer Music, 33-43, 2001.

Goodwin, M. "Adaptive Signal Models: Theory, Algorithms and Audio Applications". PhD thesis, University of California, Berkeley. 1997.

Gribonval, R., Bacry, E., Mallat,S. , Depalle, Ph. , Rodet, X. "Analysis Of Sound Signals With High Resolution Matching Pursuit ". Proc. of IEEE TFTS, 125-128. 1996.

Gribonval, R. "Approximations Non-Linéaires pour l'Analyse des Signaux Sonores". PhD thesis, Université de Paris IX Dauphine, Paris. 1999.

Gribonval,R. "Fast Matching Pursuit with a Multiscale Dictionary of Gaussian Chirps". Signal Processing, IEEE Transactions on Volume 49, Issue 5, Page(s):994 – 1001. May 2001.

Gribonval, R.; Bacry, E. "Harmonic Decompositions of Audio Signals with Matching Pursuit". Signal Processing, IEEE Transactions on, Volume 51, Issue 1:101-111. Jan. 2003.

Jones, Douglas L. e Parks, Thomas W. "Generation and Combination of Grains for Music Synthesis". Computer Music Journal, vol. 12, No. 2,27-34, 1988.

Keller, D., & Truax, B. "Ecologically-based Granular Synthesis", Proceedings of the International Computer Music Conference. Ann Arbor, MI:University of Michigan. 1998.

Kronland-Martinet, R. "The Wavelet Transform for Analysis, Synthesis, and Processing of Speech and Music Sounds". Computer Music Journal 12(4), MIT Press. 1988.

Mallat, S. and Zhang, Z. "Matching Pursuit with Time-Frequency Dictionaries". IEEE-SP, 41(12):3397-3415,1993.

Miranda, E. R., "Granular Synthesis of Sounds by Means of a Cellular Automaton", Leonardo, Vol. 28, No. 4, 1995.

Roads,C. “Introduction to Granular Synthesis”. Computer Music Journal, 12(2):27-34, 1988.

Roads, C. “Granular Synthesis of Sound”. In Roads, C., Foundations of Computer Music, Cambridge, Massachussets: MIT Press. 1987.

Truax, B. “Real-Time Granular Synthesis with a Digital Signal Processor”, Computer Music Journal, 12(2), 14-26 1988.

Truax, B. “Discovering Inner Complexity: Time Shifting and Transposition with a Real-Time Granulation Technique”, Computer Music Journal, 18(2), 38-48. 1994.

Xenakis, I. “Musiques Formelles”. La revue musicale, double numéro 253 et 254. Paris, France: Éditions Richard-Masse, 1963.

Jones, D.L. and Parks, T.W. “Generation and Combination of Grains for Music Synthesis”. Computer Music Journal, vol. 12, No. 2,27-34, 1988.

12°

Simpósio Brasileiro de Computação Musical

12th Brazilian Symposium on Computer Music

MUSIC PAPERS

e-Motion: Our Reality - 3D Motion Capture and Sonorization Via Two Cameras

Bradford Blackburn

University of Tampa, Department of Music, Tampa, FL, USA

bblackburn@ut.edu

Abstract. *This paper will present details about a two camera motion capture system that facilitates the tracking of movement in 3D. The system was developed for use in e-Motion: Our Reality, a real-time interactive music, dance, and video installation. Practical solutions to problems encountered with a two camera system are discussed. The specific sonorization techniques for dance used in the installation are also examined.*

Synopsis of the Project

e-Motion: Our Reality was an inter-disciplinary collaboration presented at the Krannert Art Museum, on the campus of the University of Illinois, Urbana-Champaign. It took place in conjunction with the "Here and Now" exhibition which featured regional artists and their work. The collaborators on this project included Bradford Blackburn (music composer), Elizabeth Johnson (dancer), Hank Kaczmarski (engineer), Ya-Ju Lin (dancer), Jessica Ray (dancer), Benjamin Schaeffer (programmer), Cho-Ying Tsai (dancer), and Luc Vanier (choreographer). Hank Kaczmarski is the Director of the Integrated Systems Laboratory at the Beckman Institute of the University of Illinois, Urbana-Champaign and coordinated the visual virtual reality portions of the project including the ten camera motion capture system, and the inclusion of 3D images of artworks being displayed in the museum. Ben Schaeffer, a research programmer for ISL, wrote the software that allowed the dancers to manipulate 3D visual imagery in real-time. Luc Vanier, a professor of dance at the University of Illinois, and a choreographer with a deep interest in motion capture technology, worked with the dancers to develop movement that tested and utilized the capabilities of both the graphic and musical virtual reality systems [Vanier, Kaczmarski, Chong, Blackburn, Williams, and de Velder 2003].

The developing choreography inspired new approaches for interfacing with the dancer's movement and thus a circle of feedback was quickly established between dancers, visual programmers, and the interactive music design. As a result, the project was continuously growing in sophistication, nuance, and organicism with each day's work. The museum visitors were able to watch this process happen up close, and view the work in progress at daily showings where they were given demonstrations of the technology and were invited to ask questions. For the collaborators this was a great opportunity to get feedback from audience members about their reactions to the technology, and to gain a greater understanding of how people interpret the various relationships between human and machine in an interactive performance.

Synopsis of the Music

The music for e-Motion was not "composed" in the traditional sense, but was instead a design for an interactive performance space that facilitated the sonorization of the dancer's. motion. Unlike previous work I have undertaken utilizing similar technology, the music for e-Motion differed because it used a three-dimensional motion capture space for the control of the music. To make this possible, it was necessary to use a minimum of two video cameras whose combined perspectives formed a 90° angle when placed on adjacent sides of the marley floor (where the dancer was performing). By overlapping the view of both cameras in this way, the dancer's movement could be viewed in any direction: front-to-back, side-to-side, up-and-down, etc., as opposed to simply looking at their motion in 2D. The output from each camera was analyzed in real-time on separate computers (logically dubbed "Computer A" and "Computer B" for their respective associations with Camera A and Camera B). The data from the analysis of each camera's output was translated into MIDI (Musical Instrument Digital Interface) data using Max (a graphic object programming language for interactive computer music) and then realized in musical sound using an external synthesizer. The sound was diffused according to the dancer's location amongst four speakers arranged quadraphonically around the motion capture space.

Video Analysis Process

The video analysis and motion tracking program which provided data concerning the dancer's movements was a third-party program written for the Max programming language called Cyclops, authored by Eric Singer [Singer 2002]. It allows for a digitized video signal to be processed in real time in a variety of ways. It works by dividing up each new frame of the digitized video into a grid of a predetermined size (for e-Motion I chose to use an 8x8 grid containing sixty-four blocks for reasons I will explain later). Within each coordinate block contained in the grid the user can include a "zone" function, which designates that a particular kind of analysis is to occur for the block containing the zone. With each successive frame the pixels contained within the block are summed together to produce an average shade or color. Depending on whether any change has occurred, and or what type of analysis has been specified for the zone, a value may be sent out. The values, along with an ounce of imagination, can then be used to control an infinite variety of processes within Max.

The 3D motion capture space for the music was that area where the separate views of the two cameras overlapped, in other words, the shared space from two different perspectives. An area where the cameras' views did not overlap, but yet was still visible to one of the cameras, was considered to be part of the peripheral 2D motion capture space.

In the arrangement used in e-Motion both cameras formed a nearly perpendicular angle with each other, therefore it is possible to think of the 3D motion capture space as similar in shape to a cube (even though its actual geometry was closer to an asymmetric polygon due to the fact that the diameter of the view seen by the video camera widened exponentially with distance from the lens), see Figure 1.

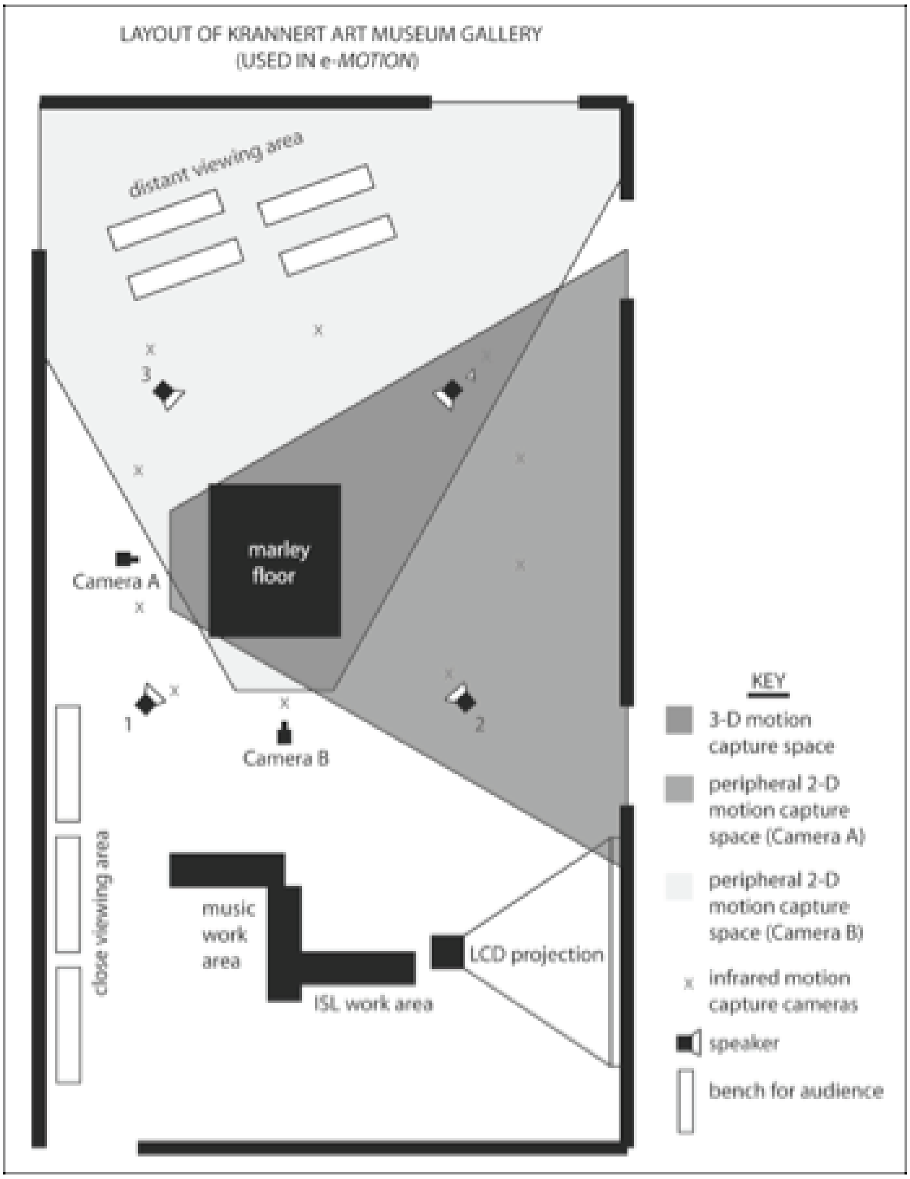


Figure 1. Camera arrangement

For each camera, the 8x8 grid within Cyclops was divided up into four quadrants of equal dimensions (4x4, or 16 blocks each). The zones assigned to each of these blocks were numbered 1 to 64 and distributed so that modular arithmetic could be used to determine which quadrant any particular zone number, that was currently registering a change in values, belonged to (quadrant 1 contained the series: 1, 5, 9...; quadrant 2 contained the series: 2, 6, 10...; quadrant 3 contained the series 3, 7, 11...; quadrant 4 contained the series: 4, 8, 12...). For example, if zone #34 was registering a change in values, than 34 would be divided by 4 to produce a remainder of 2 thus indicating that the change was occurring in quadrant 2. The quadrants for Camera A (A1, A2, A3, and A4) when combined with those of Camera B (B1, B2, B3, and B4) produced an invisible arrangement of 8 cubic sectors within the 3D motion capture space, see Figure 2.

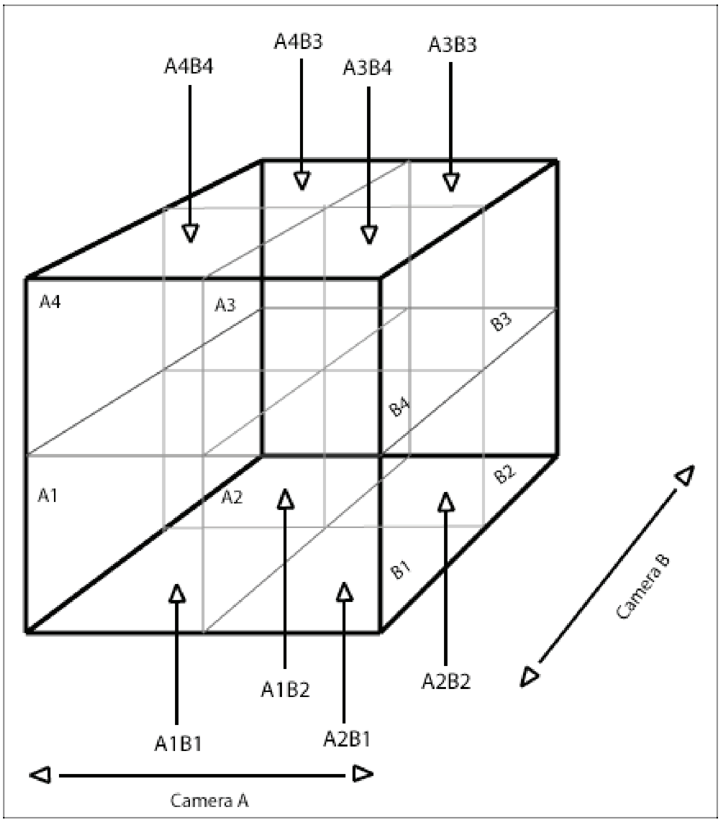


Figure 2. Camera quadrants

The central horizontal axis of both cameras was aimed at the waistline of the dancer when they were standing upright. This allowed for the dancer to isolate their control of the music between their upper and lower bodies or avoid triggering the upper sectors all together by staying below the horizontal axis'. The central vertical axis for Camera A corresponded to the division between stage left and stage right, and the central vertical axis for Camera B corresponded to the division between upstage and downstage. Since these vertical axis' along with the border of the 3D motion capture space and the peripheral 2D motion capture space were invisible to the dancer (except through sound) gaffer's tape was applied to the floor to delineate the boundary locations. The locations of the central horizontal axis' however, were left to the estimation of the dancer. By comparing the analysis from the images of both cameras it was possible to determine the dancer's general location within the eight sector cubic space (assuming they were not spiraling somewhere in the center where they might trigger all eight sectors simultaneously).

The particular kind of analysis process used for all 128 zones (total between Cameras A&B) was a difference threshold analysis on a grayscale-converted image. Whereby if a change in the average shade value for a particular block, in comparison to the value of the same block for the previous frame, exceeded a given threshold, then a +1 value would be sent out for a shade change towards the white end of the spectrum, and a -1 value would be sent out for a shade change towards the black end of the spectrum. The threshold for the process was set just high enough that only physical motion within the motion capture space would produce sufficient changes in light values that were capable of triggering an output of values from Cyclops. Using the modular operation for quadrant differentiation (previously discussed) it was possible to track the total motion occurring over a specified period of time for a particular quadrant.

The two computers shared much of the same code for kinesthetic analysis and sector differentiation. However, Computer A was allocated the additional control function of sending a synchronization pulse to itself and Computer B every two seconds. With each new pulse the total kinesthetic activity for each sector in the past two seconds was calculated. This allowed for at least a minimum of kinesthetic activity to be recorded and used as musical information, while keeping latency low enough to be relatively inconspicuous. In my experience I have found a little bit of latency actually desirable in setting music to a visual image, whether dance or film, for a variety of reasons. For one, it approximates the way we experience visual and sonic stimuli in the natural world. Another reason is that the interpretation of musical sound events is a much slower and more abstract experience compared to our instant ability to recognize visual stimuli. If the natural relationship is reversed, then the dancer would appear to be following the music and the sense of interactivity would be lost. (Enter the classic rule of effective film scoring, the orchestra swells a moment after the kiss of dramatic culmination.)

Motion to Music Interrelationships

One of the fundamental questions at the heart of an interactive performance is always how direct and palpable are the relationships between two interacting forces. How clear should the relationships be for the uninformed viewer? Naturally any answer to this question presupposes a lot about the potential audience. In the case of e-Motion the expectation was that most visitors to the exhibition would be seeing this technology for the first time, and would not have preconceived notions about what an interactive dance performance should be. It seemed likely that many would saunter through the museum at a fairly steady pace without pausing to observe any particular exhibit for very long. Since I did indeed want to make the viewer aware of the fact that the dancer had control over the music at some level, and keeping in mind my notion of who the average visitor would be, I sought to make these relationships as clear as possible and chose to establish readily observable 1:1 correspondences between the dancer and the music. In previous interactive dance performances where I have used more convoluted algorithmic processes for creating interaction, the majority of audience members have been almost entirely oblivious as to how the interaction was taking place, even when I have written extensive program notes to explain the kinds of interaction that were occurring. Which begs the question—why bother with the live interactive technology at all, when a recording of anything other than a static sine wave tone might achieve the same result through pure and simple chance? Therefore, using clearly presented connections seemed the best choice for the project.

The 1:1 correspondences that were employed included the dancer's kinesthetic motion as a control for amplitude, duration, and total note events. As the dancer's total kinesthetic motion for each of the eight quadrants was calculated every two seconds, the value from this calculation would replace the previous as the new amplitude for all note events triggered by the dancer's motion within the same quadrant. This new amplitude would only be reached one and a half seconds after it became the new value, in the intervening time there would be a gradual ramp of values to smooth out any sharp transitions that were the result of taking a tally of the kinesthetic motion every two seconds instead of in smaller increments.

Although the general term "amplitude" is being used here to refer to the acoustical loudness of the actual sound, what it really refers to in the context of the algorithms employed are MIDI attack velocity messages; these messages may control (among other things): loudness, timbre, proximity, modulation, etc. Since MIDI attack velocity messages are in a range of 128 possible values, a simple multiplication operation was used to scale the total kinesthetic motion to a usable quantity. Kinesthetic motion was also used in a similar manner to control the duration of any notes being triggered within the same quadrant. However, unlike the calculation for amplitude, duration was considered in inverse proportion to the total kinesthetic energy for the quadrant so that the more motion that occurred the shorter the durations became.

Note events were triggered by changes in zone values (the result of Cyclops detecting changes in light intensity through its difference threshold analysis) and were therefore the direct result of the dancer's motion, see Figure 3.

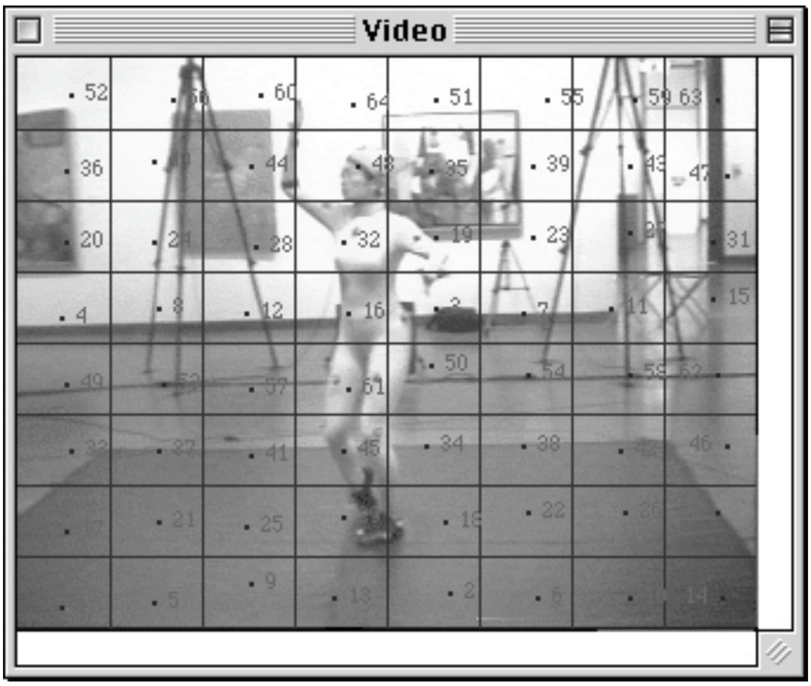


Figure 3. Grid view of dancer

By moving with isolation and poise the dancer could initiate note events very precisely, and by making larger sweeps and gestures they could create huge washes of sound. In fact, it was fascinating to hear how a conscientious dancer could sound distinctly different from an untrained mover; inevitably they sounded less random.

Additionally, the dancer's location was used to control pitch, timbre, and sound diffusion. Whenever the dancer's movement triggered a zone, the index number for the triggered zone became the MIDI note number of the note event corresponding to the change in zone values. This meant, for example, that zone #55 for Camera A (located in quadrant 3) would correspond to MIDI note 55 (unless it was transposed for esthetic purposes, as will be explained later). Assuming that the MIDI note numbers are then being filtered through a standard twelve tone equal-tempered tuning, it is already possible to predict the prevailing harmonic quality the music will have by simply looking at the distribution of zone numbers on the grid. With each quadrant being built from an integer series based on increments of four, the numbers contained within the quadrant will be mapped to members of the same augmented chord, albeit in various octaves. Furthermore, since the dancer's motion most frequently occurs as a trajectory through the same area (they don't disappear from one quadrant and reappear in another) it follows that distinct collections of augmented triads (in various register distributions) will occur and be audible. In a way, this system resembles a giant 3D pitch lattice that can be played by moving within it [Fonville 1991]. Although the preponderance of augmented chords was actually an accidental byproduct resulting from trying to find an efficient solution to the problem of locating the dancer's position in the motion capture space using a modular operation, it turned out that the resulting "neo-impressionist" sounding harmonic quality resonated in a satisfying way, esthetically speaking, with the character of the exhibition and the ambient sound of the museum itself. There was also the added bonus of facilitating palpable connections between pitch space and physical space on a perceptual level. In future projects it would be easy to circumvent this particular mapping by redistributing the zones within the grid, or creating a separate algorithm for generating pitch.

The dancer's location was also mapped to timbre in a 1:1 correspondence. Each of the eight quadrants (between Cameras A&B) were associated with a particular MIDI channel, and each MIDI channel was assigned a particular sound on the external synthesizer. With four separate audio outputs on the synthesizer the sounds were distributed in isolation to one of four speakers arranged in a quadrophonic array around the motion capture space. Each sound was also sent to a subwoofer to achieve added bass resonance. The mapping of the particular quadrants to the four speakers was done so that the dancer's location would be paralleled by the sound diffusion (via the activation of a particular timbre in a fixed location). The way it appeared to the viewer was that the sound seemed to follow the dancer through the space, and the timbre changed depending on their location.

Idio-synchro-sies

The decision to use timbres available from an external synthesizer instead of synthesizing the sound in real-time within the computer was primarily done for practical reasons. With both computers heavily tied up with the video analysis and algorithmic processes in Max, using MIDI to control outboard gear for sound generation was an efficient solution compared with adding a third computer to handle the signal processing tasks.

Within the external synthesizer (an Ensoniq TS-12) there were two sets of eight sounds each selected and programmed in advance, like a palette of colors that could be called up at will. One of the sets featured essentially familiar acoustic instruments, while the other was a hybrid of electronic and more obscure ethnic instruments. Each set produced distinctly different results, for example the electronic sounding set had sounds that would not decay automatically. This resulted in the occasional inadvertent pedal tone as the dancer tripped up the process before a note-off message could be sent to the synthesizer. The result turned out to be a desirable accident since it contrasted well with the more percussive quick-decaying acoustic sound set.

In order to create additional variety it was sometimes effective to switch one or the other cameras off. By doing so, the dancer was able to move in at least one trajectory where their movement would not trigger musical events, or would trigger them only minimally. This provided a satisfying thinning of the texture periodically, which had the effect of clearing the air.

Another method for achieving variety was to transpose the zone index numbers for one of the grids by some degree in order to shift the pitch material up or down. With extreme transpositions, there were interesting artifacts which resulted from using notes at the peripheral extremes of the synthesizer's sound sample map (eg., key noises were mapped to register extremes in some cases).

Ultimately, both these tools for variation were triggered automatically on cycles that were out of phase with each other. Computer B was given the task of toggling at random between one of the three permutations for the on/off status of Cameras A&B (1. A-on B-on, 2. A-on B-off, 3. A-off B-on) using the arbitrary time interval of 37 seconds. Computer A transposed the zone index numbers within a 128 note range at random every 51 seconds, and Computer B transposed its zone index numbers every 60 seconds. These automatic processes were allowed to run unhindered producing a gradually evolving kaleidoscope of endless possible combinations; except during the daily showings, where a manual override would be used to allow more direct control over the pacing of the performance.

Technical Challenges

One of the biggest challenges for the project was getting both computers to communicate with each other through MIDI. With a very assorted collection of gear including a MINI Macman interface, and a Tascam US-428 controller, it was possible to jerry-rig a system that seemed to work fairly well, with the occasional MIDI port overload. Every so often, when the dancer's movement became extremely active for an extended period of time, the MIDI port would choke on the flood of data causing the OMS MIDI driver application to freeze-up. This was easy to observe when children were allowed to play in the motion-capture space; with their zealous enthusiasm and unmitigated energy they proved to be among the best extreme "beta-testers" for the system. To get around this problem, a set of filtering subroutines ("speedlim" objects) were inserted into the Max algorithms to insure that the data would not overload the MIDI port.

Other problems that were encountered mostly concerned the video and its analysis. In trying to get accurate data about the dancer's location, it was crucial that the cameras were locked down, did not change their view, and were not otherwise adjusted in anyway. Once this was fairly secure, it was observed that the angle of the track lighting on the ceiling was creating long shadows from the dancer's body. To circumvent this problem, the lights were angled to be more perpendicular to the floor and focused around the center of the motion capture space. The settings within Cyclops had to be fine-tuned as well, for example, the threshold value couldn't be too sensitive or minute changes in ambient light intensity could be recorded as motion, and conversely the threshold value had to be sensitive enough to respond to subtle movements by the dancer. In order to gain increased sensitivity, the decision was made early on to switch from a 5x5 grid in Cyclops to an 8x8 grid. This proved to be a magic number since it allowed the dancer's body, when they were standing in a central location, to be divided up into enough segments to capture their movement in isolation. It also kept the processing time low and cut down on latency; and as an added bonus allowed for a direct numerical correlation to MIDI (which uses 128 as its range of values).

Observations

Perhaps one of the hardest things for dancers to get used to when working with this technology is the feeling of control that they are suddenly empowered with, since it is an aberration from the traditional relationship of music and dance. To quote one of the dancers from e-Motion, it might very well be "too much control". The default role of music in dance is to drive the dancers along or to fill a void left by the starkness of movement without words. It may seem discomforting then to have the music suddenly change from a static monolith to a malleable mirror. But it is precisely this ambiguity between having control and being surprised by the unexpected, which creates the opportunity for an authentic interactive performance. Both the dancer and the system become equal partners in this exchange with the dancers seeking to achieve greater accuracy in musical results by acquiring mechanical precision; and the composer, working vicariously through the system, seeking to undermine the regularity of the music with engineered humanist spontaneity. Like a new environment, an interactive music performance space will seem strange and exotic to the dancer when they first enter it. In a sense, the music will "play them" for as long as it takes the dancer to understand the result of their movement before its execution. Eventually though, their command of the environment will be complete and the scale will be tipped in the other direction. The moment of true interaction is the ephemeral state of equilibrium that happens in between.

References

Fonville, J., “Ben Johnston’s Extended Just Intonation: A Guide for Interpreters,” Perspectives of New Music, 1991, Vol. 29, No. 2 (Summer, 1991) pp. 106-137.

Singer, E., Cyclops (Software), 2000-2001. Commercially released in February 2002. Available from: http://www.cycling74.com/downloads/cyclops

Vanier, L, Kaczmarski, H., Chong, L., Blackburn, B., Williams, M., de Velder, A., "Connecting the Dots: The Dissection of a Live Optical Motion Capture Animation Dance Performance", New London (CT): Biennial Symposium for Arts and Technology, 2003 [cited 2009 Aug 8]. Available from: http://www.isl.uiuc.edu/Publications/final%20dance1.pdf

Winkler, T., Composing Interactive Music: Techniques and Ideas Using Max, Cambridge (MA): MIT Press, 1998.

I/VOID/O: real-time sound synthesis and video processing in an interactive installation*

Daniel Luís Barreiro[1], Sandro Canavezzi de Abreu[2], André Carlos Ponce de Leon Ferreira de Carvalho[3]

[1]Faculdade de Artes, Filosofia e Ciências Sociais
Universidade Federal de Uberlândia (UFU)
Av. João Naves de Ávila, 2121 – Bloco 1V – 38408-100 – Uberlândia – MG – Brazil

[2]Faculdade de Arquitetura, Urbanismo e Design
Universidade Federal de Uberlândia (UFU)
Av. João Naves de Ávila, 2121 – Bloco I – 38408-100 – Uberlândia – MG – Brazil

[3]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo (USP)
Av. Trabalhador São-carlense, 400 – PO Box: 668 – 13560-970 – São Carlos – SP – Brazil

dlbarreiro@gmail.com, sandroid@gmail.com, andre@icmc.usp.br

Abstract. *This paper presents some technical and aesthetic aspects involved in the conception of the interactive installation I/VOID/O concerning both the visual and sonic processes generated in real-time with Max/MSP/Jitter. It mentions the main characteristics of the installation and how the patches were implemented in order to provide coherent relationships between sound and image with the aim of offering an immersive experience for the people who visit and interact with the installation.*

1. Introduction

This paper presents real-time interactive processes with sounds and images implemented in the installation *I/VOID/O*, by Sandro Canavezzi de Abreu, with soundscapes by Daniel Barreiro. The installation was exhibited in the event *Emergência - Emoção Art.ficial 4.0*, at Itaú Cultural, Sao Paulo, from 1st of July to 15th of September 2008.

In this installation, images are captured inside a metallic sphere with a mirrored internal surface. Four cameras are used inside the sphere, one of which is placed at the tip of a stick that can be manipulated by the people who visit the installation. Two other cameras are linked to capture stereoscopic images, which are presented only in the last

* The sound synthesis implementations that are discussed here were part of a Pos-Doc research carried out with the support of Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq, Brazil.

stage of the interactive process (explanation regarding the stages, understood as levels of immersion, are presented later on in this paper). Another camera is also used to capture images inside the sphere, but not stereoscopic ones. A fifth camera is positioned on one of the walls of the installation to capture the image of the person who interacts with the sphere (see Figure 1).

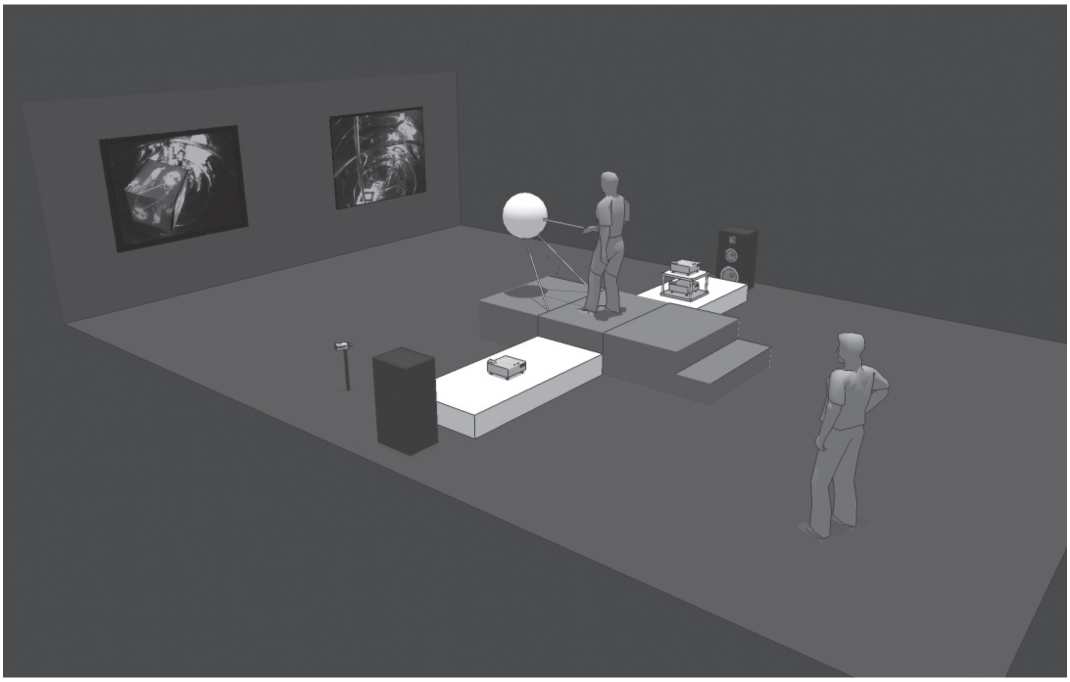


Figure 1: Diagram of the installation

The visual content of the installation is altered according to the level of immersion of the interactive process (in a total of four main levels and two transitory ones). Two video projection screens are used, placed side by side. Images, in black and white, are projected on the left screen. The right screen is only operational in the last stage (last level) of the interactive process with images in red and blue that enable 3D vision with the aid of red/blue glasses. The sounds – generated in reaction to the transformation of the images and the passage through different levels – are distributed over four loudspeakers (each one placed on one of the four walls of the installation), which contributes to generate an immersive environment.

Five computers are used in order to distribute the computational tasks and due to specific demands related to the projection of the images. They carry out the following roles:

- one computer for the real-time sound synthesis processes;

- one computer for the augmented reality system (software *artoolkit* running in Linux);

- one computer for the global management of the operations and movement tracking;

- two computers for the projection of the stereoscopic images (we decided to juxtapose the red/blue images from two different projection sources in order to obtain higher colour fidelity).

The images captured inside the metallic sphere are processed in different ways in each one of the four main and two transitory levels. The images are also analysed using various computational tools and the analysis data are sent to the computer that carries out the audio processes using the OSC protocol (Open Sound Control – see Wright et al. 2003)[1] via Ethernet connection.

The installation is seen as a system that can present different behaviours as the levels unfold. It articulates different relationships between images and sounds on each level (an example can be seen on http://br.youtube.com/watch?v=fnffoU7pX2o).

The following topics present an outline of the aesthetic proposal that informed the conception of the installation, followed by a description of the images and the immersive levels that occur in *I/VOID/O*. A following topic mentions the processes used for generating the sounds in real-time and their relationship with the images. After that, a brief evaluation of the sonic results is presented, followed by final considerations which point out to possible future developments of our work with interactive installations.

2. Aesthetic Proposal: *I/VOID/O*, a metainterface

The installation *I/VOID/O* (input/VOID/output) approaches the observation of an object as a physically interactive phenomenon. The *interactor* (person who interacts with the installation), while watching the content of a 'black-box' (this sentence is itself an oxymoron, in cybernetic terms [Wiener 1962]), interferes in such a way with the observed object that he/she ends up (re)creating it. However, that which is created, the internal reality that is observed, is nothing more than the *interface* itself. As a consequence, the interface is related to itself, a *meta-interface*, which redesigns itself continually from the initial *input* provided by the user: his/her observation.

This feedback process, which determines the non-linearity of the system, is the logical background that permeates the whole interaction. This feedback, however, is open, i.e., the feedback parameters are dynamic and are used for reorganizing the system. This dynamicity occurs as a result of the permanent confrontation between the analog and the digital domains which are present in the interface (here understood as a "field of tension" or *Schnittstelle* [Zielinski 1997]). The tension generated by this confrontation, the variation of levels of entropy in the system (in which one domain destabilizes or controls the other in an attempt to assimilate it mutually), is the material of the interface.

Light (image as reflected on the mirror and image captured by cameras) is translated in discrete parameters that change the behaviour of both sound and image in a continuous and vertiginous 'loop'. In this dynamic process, the interactor is taken to move between dimensions, or levels of immersion, touching different realities and internal universes of the 'black-box'.

This motion erodes the isolated vision of an external super-observer (as in the Classical Objectivity) and generates a cognitive friction in the observer/interactor. This

[1] http://archive.cnmat.berkeley.edu/OpenSoundControl/

friction is the result of spatial paradoxes created by the interface. These paradoxes are created by the reflection of the images on the concave mirror: they seem to release from the mirror surface and ephemerally float in the space, like a ghostly hologram. Also, counter-intuitive distortions, inversions and reversions of the images, fused in continuous visual *feedbacks* that tend to infinity, pose questions upon our understanding of the internal space of the sphere.

This cognitive friction points out to what cannot be directly observed; it points out to the shadow, the interval between dimensions – an interval that is not only void; an interval that is structural and, therefore, that organizes and supports the different dimensions[2].

I/VOID/O, therefore, approaches the observation process itself as its constitutive material. In this installation the object under observation is *observation* itself – which explains the use of several cameras and different image processing techniques that enable the appreciation of several forms of observation. The observation process incorporates *feedback* as a self-destructive process – the interface builds itself only when it destroys itself. 'Looking' is forged in order to enable observation. However, when one observes, he/she sees him/herself, and therefore stops observing, and so on and so forth. *I/VOID/O*, therefore, is about the impossibility of observation without interference. And more: it is about observation as creation and death, cyclically.

3. Images and immersive levels in *I/VOID/O*

The immersive levels in *I/VOID/O* are sequential levels that are reached and surpassed during the interactive process. Each level presents a different way of observing the interior of the sphere. The succession of levels corresponds to an increase in 'observation ability' in the manipulation of the interface.

3.1. Level I

In the first level, the images seen by the interactor are disconnected from his/her movements. The rupture of temporal linearity in the images results from the programming done in *Jitter*, which uses a video *buffer* that is updated every three seconds (in order not to overload the use of RAM). These three seconds are read randomly, i.e. the bits are not read linearly. The visual result is the temporal fragmentation of the image, which does not present the continuity that can be found in the images of movement that we see in our daily lives.

The rupture between the images and the interactor's reaction can lead him/her to seek some kind of coherence (or a more evident reactivity) by producing stronger and sudden movements[3]. When this happens, the level of entropy in the system increases and the interactor ends up trapped in this level (Level I). Entropy is understood here as

---

2 The ideas of 'super-observer' (mentioned earlier), 'cognitive friction' and 'the void between dimensions' are poetic adaptations of concepts about observation and interactivity articulated by Roessler 1998.

3 This kind of behaviour from the interactor had been observed in previous versions of the I/VOID/O, such as the one exhibited in 2005 at the Festival Internacional de Linguagem Eletrônica – FILE 2005.

'disorganised energy'. In this case, the disorganisation is the result of unbalance in the interactor's movements, which is calculated as follows: the amount of movement to the left (measured by the difference in the amount of pixels that change between two video frames, from right to left) is subtracted from the amount of movement to the right. When this difference reaches a pre-determined threshold within a certain time span (three minutes), the interactor restarts at the same level (in case he/she is in Level I) or he/she returns to the previous level.

In order to advance to another level, the interactor has to produce more controlled movements in an attempt to explore the details of the images. As a consequence, his/her movements do not disturb the system in excess and he/she advances to the next level. The deceleration of the movements can happen when the interactor starts to search for details in the image, or when he/she tries to understand the internal events inside the sphere. At the exhibition there was also an assistant who would inform the interactor about the possibility to decelerate his/her movements and the resulting reaction of the system.

It is important to mention that computer vision algorithms were used ('cv.jit' library for Max/MSP/Jitter[4]), which track the direction of movements in the images. It was necessary to add some other logical and arithmetical operations in order to quantize the variation of movement within a certain time span.

3.2. Level II

While in Level II, the interactor can notice a greater degree of coherence between his/her movements and the images that are projected on the screen. In Level I, the direction of the movements practiced by the interactor does not present any relationship with the images, due to the fragmentation of the images mentioned earlier. In Level II, on the other hand, the direction of the movements is recognised by the interactor in the images that he/she sees because the camera moves inside the sphere according to the movements that he/she makes and the images present what is captured by the camera.

However, the degree of coherence is not at its full: the concave mirror of the internal surface of the sphere generates visual paradoxes that present themselves as challenges for the understanding of the space that is being explored.

3.3. Levels III and IV

In Level III, the interactor is able to observe the interior of the sphere more accurately: images do not come from the camera located at the tip of the stick anymore. They come from another camera positioned on the internal surface of the sphere that provides a static point of observation pointing towards the centre of the sphere. It would be logical to infer that the image captured in such a way should show the stick and the camera (placed on its tip) moving inside the sphere. This, however, is not what happens. What one sees is a floating cube on the tip of the stick. On this cube, the interactor can even see him/herself, since his/her image is projected onto the surfaces of the cube (this

---

[4] For information on Max/MSP/Jitter, see http://www.cycling74.com. For information on cv.jit, see http://www.iamas.ac.jp/~jovan02/cv/

image of the interactor is captured by the camera placed on one of the walls in the space of the installation). This cube presents a phantom aspect: although it seems real, one cannot see its reflection on the internal surface of the sphere. This is due to the fact that the cube is not *really* there: it is rendered and synchronised to the stick, which gives the impression that the cube is attached to the stick. The image of the cube characterises the level of immersion related to the 'Cartesian illusion' that produces a certain degree of coherence in the internal space of the sphere. The idea of ‘Cartesian illusion’ is understood here as a construct that creates and organises a homogeneous and coherent space, which is not able to deal with congruent and parallel dimensions as understood by the topology and space of phases.

Besides the image of the cube, the interactor can occasionally visualise another perspective of the ‘Cartesian illusion’ depicted in Level IV (a transitory level): a 3D image (rendered in the form of lines) reveals distortions on the spherical space caused by the movements of the camera – the 3D grid is rendered in real-time and its vertices are continuously repositioned in relation to the intensity and the direction of the movements of the camera.

3.4. Levels V and VI

Level V is a transitory one. In this level, the interactor observes the image of the cube (rendered and synchronised to the movement of the camera) being continuously enlarged until it takes up the whole area of the projected image. This enlargement happens within six seconds and at the end of this interval, the projection is interrupted – which instantaneously activates the projection of Level VI on the right screen. Wearing red/blue glasses, the interactor can see the internal images of the sphere projected on the right screen with stereoscopic view (which provides a sense of depth to the images). When this level is surpassed, there is a return to Level I again, with images in black and white projected on the left screen.

4. The sounds in *I/VOID/O*: synthesis and sound processing in real-time

On the sonic domain, the interactive processes implemented in *I/VOID/O* are based on synthesis and sound processing techniques carried out in real-time using data from the analysis of the images and parameters that are changed randomly. Techniques of granular synthesis (see [Truax 1988], [Lippe 1994], [Keller and Rolfe 1998] and [Keller and Truax 1998]) and additive synthesis (see Dodge and Jerse 1997) are implemented in Max/MSP in a patch especially designed for the purpose of the installation.

The processes are carried out by several Max/MSP modules (*subpatches*) embedded in the main *patch* (see Figure 2). Apart from the subpatches that receive data related to the analysis of the images via OSC protocol, the main patch presents a subpatch that controls the amplitude of the sounds and their distribution over the four loudspeakers (“p volume_control”) and also another subpatch in which the synthesis and the sound processing modules can be found (“p sound_source”). Inside “p sound_source”, the modules are grouped in three different subpatches – one that generates the soundscapes for Levels I and II; another for Levels III, IV and V; and a third one that generates the soundscape for Level VI.

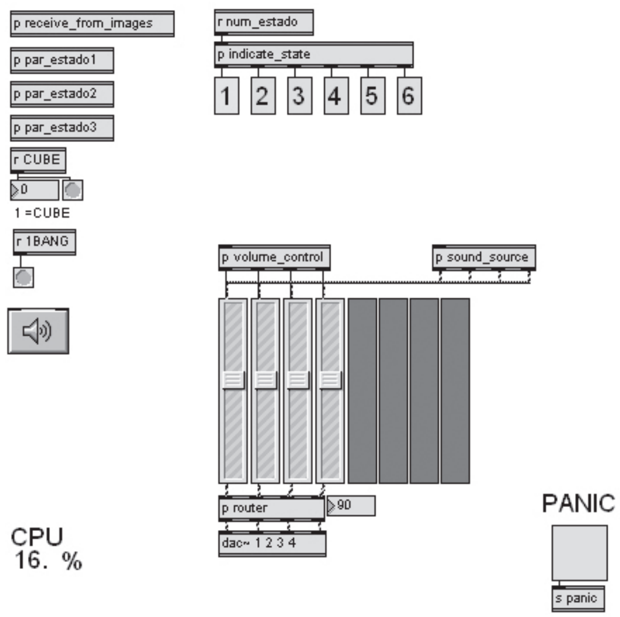


Figure 2: Audio – main *patch*

Whenever the interactor advances levels in the installation, the computer that carries out the sonic processes receives this information via OSC protocol. Changes in the visual domain are, therefore, synchronized to changes in the sonic domain.

4.1. The sounds of Levels I and II

Levels I and II are based on sounds generated by granular synthesis. Pre-selected audio files are 'sliced' in segments of very short durations (grains) and juxtaposed within time spans of variable lengths. The spectrum and the texture of the new sounds vary according to the parameters (and the variation of these parameters through time) used for the granulation process.

The most important parameters are grain size and grain rate. In the implementation of Max/MSP patch, grain size is the result of an initial grain size value added to a grain size random variation. Grain rate is also implemented as the result of an arithmetic operation involving two values defined separately – a value related to the time span between the onset of successive grains added to the result of a random variation.

In this subpatch two granulators are used in parallel, each of which generates up to 20 streams of grains from three different sound files selected beforehand. The choice of the sound files and the option for using three of them was determined empirically after trying out different possibilities and deciding for the alternative that seemed to offer the most interesting sonic results (according to the opinion of the authors). Figure 3 displays an image of one of such granulators[5].

5 These granulators were especially designed for the purpose of the *I/VOID/O* installation using features of granulation patches previously designed in Max/MSP by Erik Oña and Peter Batchelor who gave

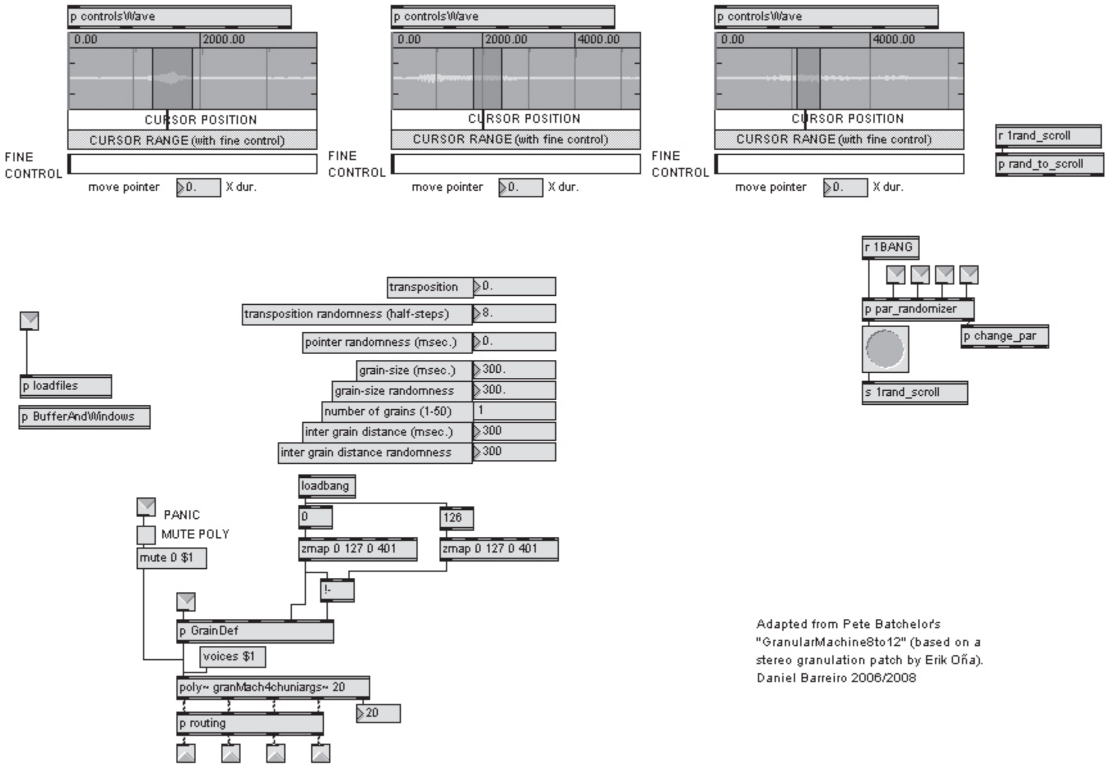


Figure 3: Granulator sub-patch

The granulation parameters change whenever there is some kind of movement – traced by the analysis of the images. The selection of the sound file to be granulated in each one of the granulators, on the other hand, is made randomly. Changes in the granulation parameters and the distribution of the sounds in the four loudspeakers are also defined by random processes within certain ranges determined beforehand. These changes occur independently in each one of the granulators. Although the sound file selection process, the changes in the granulation parameters and the distribution of the sounds in the space are determined by random operations, the resulting sonic stream is also dependant on the movements in the image. Therefore, there are some links between the interactor's actions and the sonic behaviour of the system. The random processes, however, prevent these links from being too strictly, which avoids the so called 'mickey mouse effect'.

The option for generating sounds by means of granular synthesis was mainly motivated by the morphology encountered in the sounds generated using such a technique. In Level I, the grain size and grain rate values are determined within certain ranges that prevent the system from outputting continuous sounds as the result of the granulation process. The sounds tend to present a granular character and just a few seconds of duration. Since the images in this level derive from a 3-second video *buffer* that is read randomly – in a process that shows similarities with sound granulation – both images and sounds present a non-continuous character.

---

one of the authors of this paper their patches and personal permission to use them whenever appropriate.

In Level II, the sounds generated by granular synthesis (using the process described above) pass first through a 512-band EQ implemented with Fast Fourier Transform (see Settel and Lippe 1994, 1995, 1998, 1999), and then by a reverb.

As a consequence, the passage from Level I to Level II is marked by changes in the behaviour of both the images and the sounds, as they become more continuous and closely related to the movements of the stick in comparison to the previous level. Regarding the sounds, although they are still generated by granulation, the continuous character derives from greater grain size values and smaller grain rate values. Also, the reverb applied to the sounds imposes a more continuous and resonant character to them. The sounds also present a different spectrum in comparison to the previous level, as they are changed by the EQ.

In this level, the internal space of the sphere can be more thoroughly explored – not only visually (regardless the strange forms resulted from the reflection on the curved internal surface) but also sonically by the reverberations (although synthetically produced) that happen in Level II. Also, the sounds that are generated resemble those of a metallic object – which potentially produces a connection between the sounds and the visual aspect of the sphere.

Figure 4 shows the configuration of the EQ at a certain moment in Level II (the horizontal axis represents frequency and the vertical axis represents amplitude for each of 512 bands). It can be noticed from Figure 4 that some frequency bands are completely attenuated, whereas others are reinforced in different degrees. The amplitude of each frequency band is defined randomly and set to a new value after time spans greater than 1000 miliseconds. The actual amplitude applied to each frequency band, however, does not change abruptly from one setting to the next, as the values are slowly interpolated, which is carried out by the vectral~ object in the pfft~ subpatch. Therefore, although the resonant frequencies and their amplitudes are determined by a random process, the actual changes are smooth and, therefore, a resonant sonic structure can still be obtained.

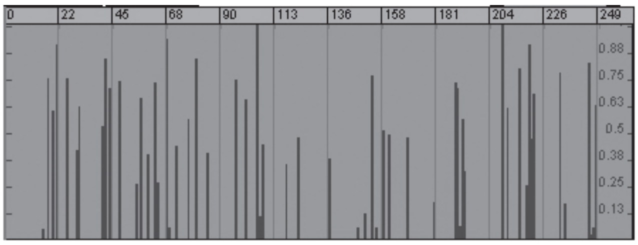


Figure 4: EQ frequency bands (FFT)

4.2. The sounds of Levels III, IV and V

In Levels III, IV and V, sounds are generated by additive synthesis (superposition of sine waves), using eight synthesis modules that superimpose six sine waves each (see in Figure 5 an image of one of these modules).

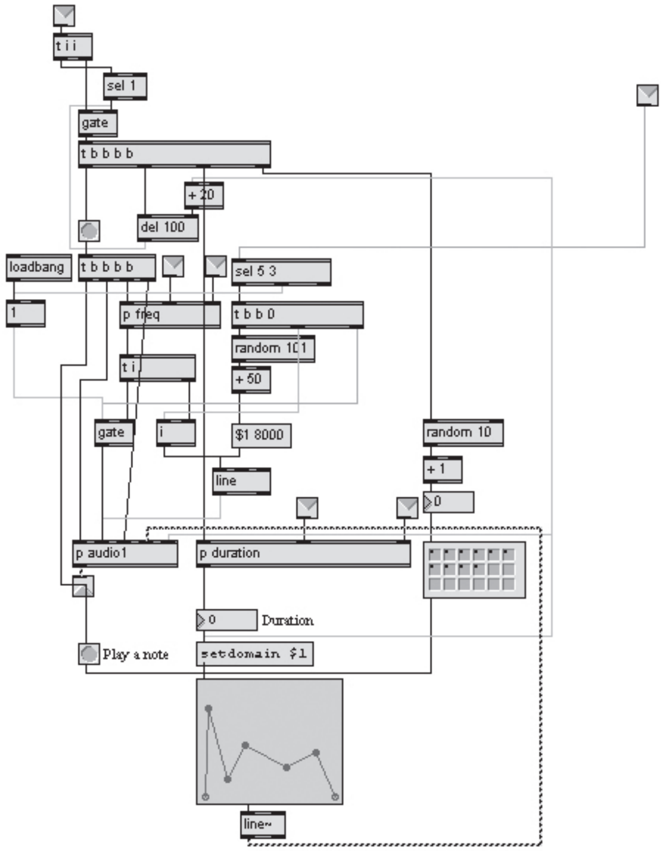


Figure 5: Additive synthesis module

The relationship between the frequencies of the six sine waves in each composed sound and their dynamic envelope are determined by the random selection of eight different *presets*.

The option for a synthesis technique other than the one used in previous levels has a connection with the change in the visual domain that happens in Level III. In this level, images are captured from a static camera pointing towards the centre of the sphere (and not the camera from the tip of the stick anymore). The form of the cube as a Cartesian object is associated with sounds obtained by means of additive synthesis.

Whenever there is a transition to Level IV (which can be visualised as lines rendered in 3D), the sounds generated by additive synthesis are subjected to slight variations using amplitude modulation.

When the interactor advances to Level V (a transitory stage in which the cube expands until it takes up the whole screen) the synthesised sounds perform a glissando towards the low register, which links the sounds of this level with the low registered sound from Level VI.

4.3. The sounds of Level VI

Level VI is based on a single long sound in the low register produced by granular synthesis. The process that was used is similar to the one described for Levels I and II, with the difference that the parameters are configured in such a way as to produce a continuous sound without a strongly noticeable granular character. This long sound rotates in the room, moving faster each time it turns around the space. After 2 minutes, both sound and image return to their configuration in Level I.

5. Considerations on the sonic results obtained

The aim of this implementation was to generate sounds in real-time that could work in a coherent and well-integrated way with the images.

Both the sounds and the images generated at a certain level present global features that happen in all occurrences of that specific level. As a consequence, the whole cycle of six levels of immersion maintains some kind of consistency in several of its occurrences.

It was possible to verify that the random processes used for varying the synthesis and processing parameters did not compromise this global sonic coherence, since they operated only in the definition of the micro-elements (details) of the sonic structures (see [Keller 2000] and [Keller and Capasso 2006]).

The synchronicity between sound and image in the passage from one level to the next and the use of analogies between the behaviour of the images and the morphology of the sounds contributed to provide coherent relationships between both, helping to create the immersive environment of the installation.

6. Final Considerations

This paper presented the main characteristics of the interactive installation *I/VOID/O* and the way synthesis and sound processing techniques were implemented in real-time in order to work with the images and the general aesthetic motivation of the installation.

Granular and additive synthesis techniques were used. The parameters of synthesis were altered using data from the analysis of the images, messages indicating the beginning of each level of immersion and also random processes.

For our future interactive projects, it would be interesting to explore the artistic potential of other computational processes and models that we have been studying, such as swarm intelligence, and the use of other kinds of interactive interfaces, such as sensors and the *wiimote*.

7. References

Dodge, C. and Jerse, T. A. Computer Music: Synthesis, Composition, and Performance. Wadsworth Publishing Company, 1997.

Keller, D. (2000). "Compositional processes from an ecological perspective". Leonardo Music Journal, 10, pp.55-60.

Keller, D. and Capasso, A. (2006). New concepts and techniques in eco-composition. Organised Sound 11(1), pp.55-62.

Keller, D. and Truax, B. (1998). "Ecologically-based granular synthesis". Proceedings of the International Computer Music Conference. Ann Arbor, MI: ICMA. http://www-ccrma.stanford.edu/~dkeller/pdf/KellerTruax98.pdf

Keller, D. and Rolfe, C. (1998). "The corner effect". Proceedings of the XII Colloquium on Musical Informatics. Gorizia: AIMI. http://www-ccrma.stanford.edu/~dkeller/pdf/KellerRolfe98.pdf

Lippe, C. (1994). "Real-Time Granular Sampling Using the IRCAM Signal Processing Workstation". Contemporary Music Review 10(2). United Kingdom: Harwood Academic Publishers, pp.149-156.

Roessler, O. E. Endophisics: The World as an Interface. World Scientific Publishing Company, 1998.

Settel, Z. and Lippe, C. (1994). "Real-Time Timbral Transformation: FFT-based Resynthesis". Contemporary Music Review, Vol.10(2). United Kingdom: Harwood Academic Publishers, pp.171-180.

Settel, Z. and Lippe, C. (1995). "Real-time Musical Applications using Frequency-domain Signal Processing". In: IEEE ASSP Workshop Proceedings. New York: Mohonk. http://ieeexplore.ieee.org/iel2/3495/10326/00482997.pdf

Settel, Z. and Lippe, C. (1998). "Real-time Frequency Domain Signal Processing on the Desktop". In: Proceedings of the ICMC1998. San Francisco: ICMA, pp.142-149.

Settel, Z. and Lippe, C. (1999). "Audio-rate control of FFT-based processing using few parameters". In: Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99), Trondheim.

Truax, B. (1988) "Real-Time Granular Synthesis with a Digital Signal Processor". In: Computer Music Journal, Vol. 12(2), pp. 14-26.

Wiener, N. Cybernetics, or Control and Communication in the animal and the Machine. Cambridge: MIT Press, 1962.

Wright, M., Freed, A. and Momeni, A. (2003). "OpenSound Control: State of the Art 2003". In: Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03). Montreal, Canada, pp.153-159.

Zielinski, Siegfried. Interfacing Realities. Rotterdam: Uitgeverij De Baile and Idea Books, 1997.

Ubiquitous Music: Concepts and Metaphors

Marcelo S. Pimenta[1], Luciano V. Flores[1], Ariadna Capasso[2], Patricia Tinajero[3], Damián Keller[2]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre, RS – Brazil

[2]Núcleo Amazônico de Pesquisa Musical – Universidade Federal do Acre (UFAC)
Caixa Postal 500 – 69.915-900 – Rio Branco, AC – Brazil

[3]School of Art – University of Tennessee (UTK)
1715 Volunteer Blvd. – 37996-2410 – Knoxville, TN – USA

{mpimenta,lvflores}@inf.ufrgs.br, ariadna102@hotmail.com, ptinajer@utk.edu, dkeller@ccrma.stanford.edu

Abstract. *Ubiquitous Music is a new area of research that encompasses ubiquitous computing, mobile and networked music, eco-composition and cooperative composition. This article examines both the metaphors for interaction and the musical activities that can be supported by ubiquitous music systems. Music making is characterized as an activity involving pragmatic-epistemic actions constrained by natural and social affordances. Music composition – a predominantly epistemic activity – results from the interactions between the musician's personal environment and the ecological niche where the activity takes place. Thus, the interaction metaphors encompass agents, tools, environment, and activities, providing a conceptual and methodological framework for musical and computational developments in ubiquitous music research. An example of a ubiquitous music work is included: Green Canopy, On the Road.*

1. Introduction

Ubiquitous music systems can be defined as musical computing environments that support multiple users, devices, sound sources and activities in an integrated way. Regarding technology, at the least, ubiquitous music systems should support: mobility, social interaction, device independence, and context awareness.

Our work stands at the intersection of mobile and networked music with ubiquitous computing technology and concepts [Weiser 1991], involving open, participative, non-trivial musical practices. Previous work on interactive installations, performance art, eco-composition, and cooperative composition partially fit within the concept of ubiquitous music. Although this term has recently appeared in the literature [Holmquist 2005; Holmquist and Tanaka 2005], there has not been any attempt to define a workable methodology that contemplates both the musical and computational issues raised by these practices. In order to establish a suitable theoretical framework for experimentation and artistic development, we will address both the categories of musical activities that can be supported by ubiquitous music systems, and the metaphors for interaction that can be applied to their design.

2. Ubiquitous Music: Activities

As composers, music practitioners, and system designers, we believe that ubiquitous music systems design should be guided by the requirements of music making. So, before tackling specific methodological issues we need to answer a basic question: What is music making? Or more specifically, what does music-making involve in the context of ubiquitous systems?

2.1. Activity and Affordances

For Leont'ev (1978), activity is at the center of human life: as soon as an activity is completed it is replaced by another activity. He suggests that activity has a circular structure: "initial afferent and effector processes regulating contacts with the objective environment, then correction and enrichment by means of reverse connections of the original afferent image" [Leont'ev 1978: 53]. This circular connection between physical and perceptual processes is a key characteristic of Gibsonian approaches [Chemero and Turvey 2007; Gibson 1966]. Thus, a research agenda that brings together Activity Theory and Ecological Psychology provides a firm ground for the study of human activity [Baerentsen and Trettvik 2002].

Our perception of the environment is shaped by the constant interactions with the objects and beings that surround us. These interactions are constrained by the possible actions that can be exerted upon the objects, that is, by their natural affordances [Gibson 1979: 127]. The permanent cycle "action / perception / attunement / new action" is at the core of the process of adaptation to a new environment. Or, more accurately stated, the mutual adjustment between environment and individual can be observed through the set of affordances that emerge from this process. So affordances can be defined either as being properties of the environment that are actualized by the agents' actions, or as relational properties of agent-environment systems [Chemero and Turvey 2007].

The basis for the perception of affordances is the temporally extended perceptual activity [Baerentsen and Trettvik 2002]. Affordances exist at the moment the organism interacts with the environment through structured actions. Perceptual activity, i.e. the efferent commands to muscles to establish contact with objects, and the influence of the perceived objects on the activity, via afferent feedback, inform the organism about the changes in the environment. Thus, affordances and activity are inextricably interrelated. More specifically, affordances are features of activity systems that include the physical environment and the organism's phylogenetic characteristics transmitted through the generations as species-specific adaptations to the ecological niches.

At a finer level of description, Leont'ev (1978) establishes a distinction between actions and activities. "When a concrete process is taking place before us, external or internal, from the point of view of its relation to motive, it appears as human activity, but when it is subordinated to purpose, it appears as an action or cumulation of a chain of actions" [Leont'ev 1978: 64]. Within the realm of physical/digital systems, Kirsh and Maglio (1994) propose two types of actions: epistemic and pragmatic. From their perspective, pragmatic actions serve a single function: to change the world. On the other hand, epistemic actions simplify the problem-solving task by uncovering hidden information and bringing the agent closer to its goal.

From an Activity Theory perspective [Leont'ev 1978: 68], internal activity that serves a cognitive motive is carried out through external actions or motor operations.

Similarly, the actions and operations that realize external activity may constitute internal – cognitive-physiological – processes, but they always keep their integrity as actions or operations. Therefore, a separation between cognitive and motor actions is unwarranted; thus, we could think of psycho-motor actions as part of a feedback process that involves both epistemic and pragmatic activities.

Generally, we could say that the organism-environment system is just a collection of affordances. Because these affordances are dependent upon the agent's personal history of interactions, we have to restrict our definition of ecological niches to the specific relationship agent-environment, that is, to the personal environment [Keller and Berger 2001] or the personal sense [Leont'ev 1978].

2.2. Social Affordances and Musical Activities

The unit of analysis in studying human mediated activity is an activity system, a community of actors who have a common purpose [Miettinen 1997]. Social mediatedness is characterized by constraints mediating the interaction between the individuals within the activity system. The focus of study moves away from isolated subjects to encompass the interaction between the individual, the artifacts and the other individuals in a dynamic changing environment. Thus, the collective activity system connects the psychological, the cultural and the ecological niches where activity takes place.

All organisms exert pressures on their habitats modifying the environment to suit their needs. In the case of the human species, these processes guide the development of tools within the context of cultural societal praxis. In other words, cultural activities involve interactions with artificial habitats and with other organisms and these interactions are constrained by canonical or social affordances [Costall 1995]. This specific type of affordances regulates community exchanges and fosters the development of physical tools to fulfill specific societal needs.

As Christopher Small (1998) suggested, musical practices are only part of a unified system of social interactions. If music is understood as social activity, the tools and concepts developed in Activity Theory can be applied to musical research. From this epistemological perspective, we can analyze musical activities as comprising systems of epistemic-pragmatic actions with specific goals. In turn, these goals will guide the implementation requirements of systems that support musical activities.

Acoustic musical instruments are just one example of tools that emerged out of a process shaped both by the environmental and societal pressures. Musical instruments co-evolved with musical practices on the one hand constraining the ability of musicians to establish new forms of sonic organization, and, on the other, providing opportunities for novel forms of music making. For instance, orchestral acoustic instruments – through their specific set of affordances – came to be suited for music thought for a single player per instrument, playing inside a concert hall.

But social affordances not only influence the development of tools, they also provide a context for the application of conceptual frameworks. Compositional paradigms are just one form of social affordance. They serve as an interface between the sonic potentialities of objects (natural affordances) and the common musical knowledge shared by the members of a society (just for the record, this knowledge also belongs to the realm of social affordances). Thus, compositional systems that are well-adapted to the available physical tools and that fulfill the current societal needs are the ones that survive social and environmental pressures.

2.3. Compositional Activities

Composition requires the exploration of numerous possible outcomes, involving tasks such as categorization, organization and planning, among others. More precisely, we could say that composition involves pragmatic-epistemic actions with the goal of internalizing micro, meso and macro imagery. From an eco-compositional perspective [Keller and Capasso 2006], the three space-time competencies required by compositional activities can be defined in the following terms:

a. Micro space-time imagery: involves the prediction of processes applied upon structural and transformational invariants.

- Structural invariants: describe the sonic qualities of events within a static spatial configuration.
- Transformational invariants: inform about sonic qualities of events within dynamic spatial configurations.

b. Meso space-time imagery: establishes the outcome of processes at a meso-time level, i.e., taking into account variables such as phase, density and distribution of meso-time processes.

c. Macro space-time imagery: involves the prediction of perceptual relationships among sonic events and across multiple time levels.

Micro space-time imagery encompasses the behavior of sound sources such as musical instruments, resonant objects, synthesis algorithms, etc. On the transformational side, it demands the prediction of outcomes resulting from DSP processing, spatialization and other types of manipulations of sonic material. Macro space-time imagery – comprising the perceptual relationships among sonic events across multiple temporal levels – is constrained by short-term and long-term human memory limitations and by selective attention processes.

Given that the personal environment results from the history of interactions between the individual and the ecological niches where the activities take place and that this process is constrained by the social affordances, the composer cannot predict whether his imagery will match the listener's. Nevertheless, two tentative strategies may be used to partially close the gap between the composer's and the listeners' imagery: 1) to implement artificial eco-niches that are as much removed from any pre-existing niche as possible; 2) to exploit natural affordances that are common to most human environments. Eco-composition usually takes the second approach.

3. Ubiquitous Music: Metaphors for Interaction

So far, we have characterized music making as a human activity involving pragmatic-epistemic actions which are constrained by natural and social affordances. We have also conceptualized music composition as a predominantly epistemic activity that results from the interactions between the musician's personal environment and the ecological niche where the activity takes place. This section will address how musical interaction metaphors impact the design of ubiquitous music systems.

A classic interface metaphor, the WIMP model uses elements such as menus, dialog boxes and scrollbars to act as mediators between users' actions and the objects being manipulated [Beaudouin-Lafon 2000]. The advantages of the WIMP model are its low cost and wide availability. Most interactive music systems support WIMP actions. And many standard audio procedures such as editing and mixing are usually done through mouse actions. Nevertheless, the WIMP metaphor presents several limitations:

- Users have a limited sense of engagement because their actions are achieved through the mediation of intermediate software objects [Beaudouin-Lafon 2000];
- PCs, monitors and QWERTY keyboards are usually associated with office work [Zicarelli 1991];
- Actions done with the mouse (without haptic feedback) are not an option to users with visual disabilities; and
- The standard mouse only gives access to simultaneous control of two continuous parameters.

As Beaudouin-Lafon pointed out (2004), WIMP interfaces have already reached their limits. These limits are particularly acute in the context of pervasive computing: the amount of information each individual user deals with has grown exponentially; the distribution of this information needs to be deployed over multiple computers and devices, including mainframes, desktop computers, laptops, PDAs, mobile phones and custom hardware; and the range of computer users has expanded drastically, incorporating novices to what was previously regarded as the exclusive realm of experts (music making is a particularly good example). So let us discuss metaphors better suited for musical interaction in the context of ubiquitous musical activities.

3.1. The Instrumental Metaphor

In contrast with other areas of computer science, most research in computer music systems has adopted the musical instrument as the ideal metaphor of interaction [Wandeley and Orio 2002; Wessel and Wright 2002]. For music performance done in real-time, this type of metaphor is the one that has the longest cumulative knowledge. Performance activities demand low-latency, immediate sonic outcome, making the "one-gesture-to-one-acoustic-result" [Wessel and Wright 2002] the ideal benchmark for implementation and testing. On the other hand, creative, exploratory activities such as the compositional activities discussed in section 2.3, are not readily supported by this paradigm.

Interfaces developed following the instrumental metaphor foster musical activities tailored after the performance of acoustic instruments. Within this context, Wanderley and Orio (2002) define a musical performance as the continuous changes of sound parameters exerted by a controller. Regarding the physical/digital mapping, music system designers should account for the perceived relationship between gestures and changes in the performance parameters and the level at which these features can be controlled. The accuracy, resolution, and range of perceived features should be determined, the focus being what the user perceives rather than the actual values of the control parameters.

The performance of a musical instrument requires very precise timing. Wanderley and Orio (2002) propose that musical tasks should strive to attain temporal precision so that musicians have complete temporal control of the performance parameters.

Transposing this guideline to multiple-user task management means prioritizing real-time strategies for synchronous control of multiple parameters on multiple devices.

The instrumental approach introduces a difficult problem for ubiquitous music systems. As Barbosa (2006) has pointed out, latency in large network systems will remain in the perceivable range for the next few years. Therefore, music systems need to take this limitation into account in order to reduce the effects of network time delays. In contrast with the instrumental paradigm, Barbosa suggests conducting the general direction of musical behavior instead of producing sonic events by direct manipulation of physical controllers.

Ubiquitous music systems place further demands on the interface which cannot be fulfilled by the instrumental paradigm. A good example is the adoption of mobile devices as musical interfaces. Multiple users need to have access to the state of the system and the location where the action takes place. This demands context awareness mechanisms and location-specific configuration of parameters. Depending on the context, devices may provide sensor or actuator capabilities to the system. Thus, the instrument metaphor is necessarily broken. In the context of ubiquitous systems, a device is not a passive object that a musician can play. It is an agent in a dynamical system that adapts itself to the musical activity, to the local environment and to the other agents that interact with it.

3.2. The Cup Metaphor

A promising paradigm for ubiquitous music interfaces has already appeared in the context of multimedia performance and installation works. Several works make use of space as an unbounded, unobtrusive interface that may be freely explored by the participants. Dannenberg and co-authors (2003: 1) use an interesting metaphor to describe this situation: “the space within an empty cup is what makes the cup useful and necessary”. Thus, we could gather all these works under a common denominator: the cup metaphor.

A good example of this approach is The Urban Corridor, an interactive multimedia installation premiered in 2001 at the CU Art Galleries, Boulder [Keller et al. 2001; Keller et al. 2002]. The installation space was constructed as a corridor featuring lights, motion sensors, two slide projectors, a video projector, and a multichannel sound system.

The visual and sonic elements in The Urban Corridor consist of two layers of material: active and passive [Keller et al. 2002]. Active elements or events are triggered by the presence of the public and passive elements, or the environment, provide a constant background that reinforces the sensation of a surrounding urban landscape.

The active layer is controlled by means of four motion sensors placed at each section of the corridor to detect the presence of people. When a sensor detects motion, it sends a command to a radio frequency receiver plugged into the power line. This receiver routes the signal through the line to an interface that decodes it as a serial message.

At the heart of the system, the control software triggers both sonic and visual events. Sound events are stored as audio tracks and are played back by two CD-ROM drives. Visual events are produced by two slide projectors and three sets of lights. When motion is detected in region one, an ON command is sent to address A1, corresponding to the projector placed at the entrance of the corridor. In turn, a two-way interface translates the serial message and routes it to the power line. The projector is plugged into a module which is set to address A1. When the module receives the ON command, it turns the slide projector on.

The Urban Corridor provides a detailed example of one of the first installations to make use of the cup metaphor. The system reacts to the presence and the actions of the participants without requiring any musical expertise. Given the visual and tactile elements, the global multi-sensory experience encourages multiple forms of interaction. Accumulation of sonic material, prompted by the actions of the visitors, determines the dynamic of the piece. And by sharing the same space, participants not only relate to the artwork, they also share a common playground.

3.3. The Ecological Metaphor

Generally speaking, ecological models represent forms of interaction between agents and objects which occur along three dimensions: time, energy and space. Each axis is determined by *n* dimensions that do not necessarily represent linear or continuous mappings. Time, the first dimension, is mapped onto finite segments called events. These events are shaped by patterns of interaction between agents and objects. The processes that shape these patterns take place at three temporal levels simultaneously: micro, meso and macro [Keller 1999].

The temporal evolution of a sound event is defined by dynamic interactions between two processes: excitation and damping. This process establishes temporal constraints on the parameter range of the excitation pattern. In other words, every event starts from zero energy and builds up at an ecologically-bound rate, until the energy input stops. At this point, the damping process kicks in reducing the energy level until zero is reached. Thus, the excitation and damping processes shape the event's energy profile. By means of a single control parameter, this algorithmic structure generates ecologically constrained meso-patterns.

The sound event is effected by the dissipation of energy by an agent on an object through their natural affordances. Each event constitutes a unique instance, temporally finite and spatially localized. As long as the events can be perceptually recognized as the result of a specific interaction between an agent and an object, they are classified as belonging to a single sound class. Complementarily, a stable form of interaction between the agent and the object is usually described as a sound source. Thus, a sound class is a collection of events that share the same source.

The second dimension of ecological models – energy – is the result of complex interactions between excitation and damping processes. These processes determine how energy gets into the resonant system and how it is dissipated. The type of excitation, the state of the object, and the forms of interaction among excitation and resonance systems give shape to events. Generally, correlations and constraints on variable ranges within finite time segments approximate the behavior of real-world sound producing processes. In ecological parlance, these constraints are encapsulated in a single concept: natural affordance.

The usual representation of the third dimension of ecological models – space – consists of three axes: azimuth, elevation and distance. Nevertheless, if events are to abide by ecological rules, arbitrary mappings of temporal patterns are not possible. The sonic field is the spatio-temporal distribution of sound events produced by actions constrained by natural affordances within a spatial and temporal horizon. The limitations are not only determined by physics but by the available modes of interaction between agents and objects within the specific ecological niche. In other words, the behavior of agents and objects is constrained by their natural affordances producing events which are limited to a spatio-temporal horizon.

4. Green Canopy

Green Canopy is a series of sculptural sound installations involving elements and sound inspired and collected in the North-Western Amazonian rainforest (Green Canopy: The Tree, Green Canopy: The Forest, Green Canopy: The Bud) [Keller et al. 2005, 2006]. The sculptural elements of the work are built entirely from recycled materials, including PVC pipes, carpet padding and crocheted plastic bags. Green Canopy has been featured in exhibits at Sculpture Space (Utica, NY, 2005), the 6th Kingston Sculpture Biennial (Kingston, NY, 2005), Hamilton College (Clinton, NY, 2006), LMAKprojects (Williamsburg, NY, 2006), the Islip Art Museum (East Islip, NY, 2006), MACO (Mexico City, 2006), and the Preview Berlin Art Fair (Berlin, June 2006).

All previous versions of Green Canopy enforced a dynamic group interaction. In the case of The Tree people would walk toward or away from the sculpture, experiencing vertical and horizontal phase relationships among sound sources (Figure 1). The Forest made use of a sonic environment that surrounded all participants sharing the common space (Figure 2). The Bud only allowed for a limited number of people to stand close to the sculpture, thus demanding alternation between groups to listen to the work.

Figure 1. Green Canopy: The Tree

Figure 2. Green Canopy: The Forest

Green Canopy, On The Road is the latest implementation within this series. In this version we further expand an issue previously explored in Green Canopy: The Bud: portability. On the Road opens up its sonic material to exploration by extending a concept borrowed from HCI techniques: the music probe [Gaver et al. 1999]. Music probes are devices designed to let users establish their personal experience of a musical work. While serving as a framework for art experimentation, they provide data for developers to refine their decisions on architecture and interface design. The probe works as a sensor / transducer system, allowing the collection of data at the site of interaction. Channels of interaction include sound and movement. Two variables that influence the usability of the system within the context of compositional activities are studied: the ability to manipulate the temporal relationship among sound events and the perceptual limits on the number and characteristics of the samples used. On the Road uses the music probe infrastructure to give users the ability to mix their own version of the work. Because the probe has been implemented for a portable device, listeners can carry the work with them. Thus the type of experience provided by this version of Green Canopy is mostly individual and self-contained.

5. Final Discussion

Ubiquitous Music, an emergent research field that integrates computer music and ubiquitous computing, presents exciting new challenges and possibilities for music making. This paper has focused on key issues for ubiquitous music system design, providing a conceptual and methodological framework for future developments. In line with broad approaches to Human-Computer Interaction [Bevan 1995], we have not dealt with specific system details and have avoided techniques that restrict the applicability of the proposed framework. As research moves on, user demands will dictate the needs for development within narrower contexts.

Adamczyk and collaborators (2007) ask "how might public presentation and communication of highly situated HCI new media projects be made compelling to new audiences?". Ubiquitous music systems may provide part of the answer. By embedding musical tools in everyday consumer devices, non-musicians are given a chance to participate in a growing community of music practitioners. Placing music-making as an extension of everyday activities reduces the cognitive cost of several years of training with non-intuitive interfaces and the highly specialized knowledge of the common practice musical syntax.

By applying the metaphors described in this paper, under a broad HCI perspective, we pave the way to a wide range of possibilities regarding the use of multiple devices as musical interfaces, from the control of notes and continuous sound parameters (within the instrumental paradigm) to the emergent properties of social collective actions in artistic spaces (within the cup paradigm). We believe that in these various contexts (particularly when actions are not explicit and are based on mundane, everyday activity), the user gains intuitive control over relevant musical parameters. Thus, we may empower both musicians and non-musicians to express themselves in collective, open-ended music making.

The ability to adapt to context through awareness of environmental variables – a key requirement of ubiquitous systems – changes the basic design philosophy. New media audiences should not have to deal with generic musical instruments that need to be mastered in order to make sound. Participants of ubiquitous new media works only need to be concerned with the creative aspects of the artistic experience: exploration and experimentation of forms and content. It is the system – and not the user – the one that should adapt its behavior to each specific context.

From an eco-compositional perspective, acknowledging the existence of natural and social affordances has a clear corollary: we cannot separate agents from objects, tools from activities, and actions from locations. What we construct, as musicians, are ecological niches or habitats where musical activities can exist. Depending on the characteristics of these eco-niches, including the agents (users), tools (systems), environment (location, space), and activities (performance, composition), we define a specific set of forms of interaction (affordances). As music systems developers we design interfaces that support interaction and co-adaptation between agents and environment, the sonic result being just a by-product of this process.

Sound Examples: Green Canopy, On the Road – fragment.

Acknowledgments

This work is being partially supported by the Brazilian research funding agencies CNPq (grants to Marcelo Soares Pimenta, and grants 301982/2008-2, 478092/2008-3, and 571443/2008-7 to Damián Keller) and CAPES (grant to Luciano Vargas Flores). Green Canopy was made possible by the generous support of Sculpture Space, ETM Solar Works, Golden Artist Colors, Inc., the Division of Composition Studies, College of Music, University of North Texas, the Islip Art Museum, and the School of Art, University of Tennessee.

References

Adamczyk, P. D., Twidale, M. B., Hamilton, K. and Bailey, B. P. (2007) "HCI and New Media Arts: Methodology and Evaluation", Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2007). San Jose, CA: ACM, p. 2813–16.

Baerentsen, K. and Trettvik, J. (2002) "An Activity Theory Approach to Affordance", Second Nordic Conference on Human-Computer Interaction (NORDI-CHI 2002). Arhus: p. 51–60.

Barbosa, A. (2006) "Displaced Soundscapes: Computer Suported Cooperative Work for Music Applications." (PhD Thesis). Pompeu Fabra University. Music Technology Group.

Beaudouin-Lafon, M. (2000) "Instrumental Interaction: An Interaction Model for Designing Post WIMP User Interfaces", Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2000). New York: ACM, p. 446–53.

Beaudouin-Lafon, M. (2004) "Designing interaction, not interfaces", Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04). New York: ACM, p. 15–22.

Bevan, N. (1995) "Usability is Quality of Use", Proc. of the VI International Conference on Human-Computer Interaction. Amsterdam: Elsevier, p. 349–54.

Chemero, A. and Turvey, M. T. (2007) "Gibsonian Affordances for Roboticists", Adaptive Behavior 15(4): 473–80.

Costall, A. (1995) "Socializing Affordances", Theory and Psychology 5: 467–81.

Dannenberg, R. B. et al. (2003) "Sound Synthesis from Video, Wearable Lights, and 'The Watercourse Way'", Proc. of the 9th Biennial Symposium on Arts and Technology. New London, CT: Ammerman Center for Arts and Technology, p. 38–44.

Gaver, B., Dunne, T. and Pacenti, E. (1999) "Cultural Probes", Interactions, January-February: 21–29.

Gibson, J. J. (1966) "The Senses Considered as Perceptual Systems." Boston: Houghton Mifflin.

Gibson, J. J. (1979) "The Ecological Approach to Visual Perception." Hillsdale: Lawrence Erlbaum.

Holmquist, L. E. (2005) "On the Edge: Ubiquitous Music", Interactions 12(4): 71, 78.

Holmquist, L. E. and Tanaka, A. (2005) "Ubiquitous Music: How are Sharing, Copyright, and Really Cool Technology Changing the Roles of the Artist and the Audience?", SIGGRAPH 2005. Los Angeles.

Keller, D. (1999) "touch'n'go: Ecological Models in Composition." (Master of Fine Arts). Simon Fraser University, Burnaby, BC.

Keller, D. and Berger, J. (2001) "Everyday sounds: synthesis parameters and perceptual correlates", Proc. of the 8th Brazilian Symposium on Computer Music. Fortaleza, CE, Brazil: SBC.

Keller, D. and Capasso, A. (2006) "New Concepts and Techniques in Eco-composition", Organised Sound 11(1): 55–62.

Keller, D., Capasso, A. and Tinajero, P. (2001) "The Urban Corridor [Interactive Installation]." Boulder, CO: CU Art Galleries.

Keller, D., Capasso, A. and Tinajero, P. (2005) "Green Canopy: The Tree [Sound Sculpture]", 6th Kingston Sculpture Biennial. Kingston, NY.

Keller, D., Capasso, A. and Tinajero, P. (2006) "Green Canopy: The Forest [Sound Sculpture]." Clinton, NY: Hamilton College.

Keller, D., Capasso, A. and Wilson, R. S. (2002) "Urban Corridor: Accumulation and Interaction as Form-bearing Processes", Proc. of ICMC 2002. Göteborg: ICMA.

Kirsh, D. and Maglio, P. (1994) "On Distinguishing Epistemic from Pragmatic Action", Cognitive Science 18: 513–49.

Leont'ev, A. N. (1978) "Activity, Consciousness, Personality." New Jersey, NY: Prentice Hall.

Miettinen, R. (1997) "The concept of activity in the analysis of heterogeneous networks in innovation process", CSTT Workshop 'Actor Network and After'.

Small, C. (1998) "Musicking." Wesleyan University Press.

Wanderley, M. M. and Orio, N. (2002) "Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI", Computer Music Journal 26(3): 62–76.

Weiser, M. (1991) "The Computer for the Twenty-First Century", Scientific American 265(3): 94–101.

Wessel D. and Wright, M. (2002) "Problems and Prospects for Intimate Musical Control of Computers", Computer Music Journal 26(3): 11–22.

Zicarelli, D. (1991) "Music for Mind and Body", Electronic Musician 7(2): 154.

A Música no Cinema Mudo e o Instrumento Musical Digital

Pedro Patrício

Doutorando em Ciência e Tecnologia das Artes, especialização em Informática Musical, CITAR, Escola das Artes, Universidade Católica Portuguesa (UCP), Porto, Portugal

pp2007.pt@gmail.com

http://pp2007.pt.googlepages.com

Resumo. *Seguindo o exemplo das exibições musicais realizadas ao vivo durante a projecção de filmes mudos e por influência da descoberta de várias máquinas utilizadas quer para gerar efeitos sonoros quer para musicar filmes mudos, construímos recorrendo às novas tecnologias e à informática musical, um instrumento musical digital. Para isso, utilizou-se um dispositivo electrónico com* WiFi (iPod touch), *uma aplicação que permite a comunicação OpenSound Control (OSCRemote) e uma plataforma multimédia de programação por objectos (Pure Data). No final do artigo, fazemos algumas considerações estéticas sobre as opções musicais utilizadas.*

Abstract. *This work discusses the repertoire, the mechanical musical instrument and the paper attributed to music in the beginning of the silence movies. We constructed, appealing to the new technologies and to musical computer science, a digital musical instrument that could potentially play the same function as some machines used to generate sound effects and to play music in silence movies. In order to do so, we used a device with WiFi (iPod touch), an application that uses OpenSound Control (OSCRemote) for communication and a multimedia-programming platform (Pure Data). In the end of the article, we present some aesthetic considerations about the musical options that we made.*

1. Introdução

Foi-nos proposto, no âmbito do 6º Festival *Black&White em* 2009, sonorizar ou musicar ao vivo 3 curtos excertos de vídeo extraídos de uma curta metragem realizada por Manoel de Oliveira intitulada *Douro, Faina Fluvial* (1931)[1]. O concerto aconteceu na Cerimónia de Abertura do referido festival, no Auditório Ilídio Pinho - Universidade Católica Portuguesa (Porto)[2]. Os 3 excertos de vídeo com a duração aproximada de 4 minutos cada eram constituídos por um excerto do filme original, um *remake* do

[1] Filmado em 35 mm, preto e branco e com a duração de 21 minutos. O filme é influenciado pela estética vanguardista do documentário soviético praticada por Dziga Vertov.

[2] "(De)criando à Sombra de Oliveira", Formação Variável de Laptops do Citar (Piano: Telmo Marques; Difusão Sonora: José Luís Ferreira; Laptops: André Rangel, João Cordeiro, Miguel Cardoso, Pedro Patrício, Ricardo Guerreiro e Vítor Joaquim).

primeiro excerto e por um filme fantasma (sobreposição dos dois primeiros excertos). Para esse fim, a nossa participação, para além da intervenção ao vivo (processamento de som captado em directo e manuseamento de laptop), consistiu na concepção e construção de um instrumento musical digital utilizando o *iPod touch*[3] e que é descrito nos parágrafos seguintes.

1.1. Cinema Mudo: repertório e instrumentos musicais mecânicos

Ao falar de música para cinema é necessário referir as exibições que aconteciam ao vivo durante as projecções dos filmes mudos. A primeira utilização conhecida da música ao vivo no cinema ocorreu em 28 de Dezembro de 1895, quando a família Lumière projectou no Grand Café, no Boulevard de Capucines em Paris, os seus filmes acompanhados por um piano. Durante o desenvolvimento do cinema, os produtores pretendiam atribuir a cada filme a sua própria música, e para isso faziam encomendas a compositores para comporem especificamente para um determinado filme. Na história da música ocidental, um dos primeiros compositores a compor música especificamente para cinema, foi Camille Saint Saens. Compôs a música para o filme de Henri Lavedan, *L'Assassinat du duc de Guise* (1908). Contudo, esta ideia não foi muito disseminada, por representar um custo adicional ao orçamento final dos filmes. Como alternativa, por volta de 1913, orquestras e pianistas recorriam a catálogos musicais especiais. O mais conhecido exemplo é o *Kinobibliotek* ou *Kinothek* de Giuseppe Becce, publicado em Berlim em 1919. As peças musicais do catálogo eram registadas de acordo com os seus estilos e a carga emocional que presumivelmente elas provocariam ao serem ouvidas. Era recorrente pensar-se que dentro da música clássica existia tanta abundância de peças, que estando elas divididas em categorias em relação ao catálogo de Becce, havia música praticamente pronta para qualquer cena em qualquer tipo de filme. Assim, obras de Beethoven, Mozart, Grieg, Bach, Verdi, Bizet, Tchaikovsky, Wagner e, em geral, qualquer uma que não fosse protegida pelas leis do direito de autor, eram frequentemente utilizadas. Para os teatros menores, que não tinham nem dinheiro nem espaço para uma orquestra, foram inventadas várias máquinas no sentido de substituir as orquestras. Estas máquinas apareceram primeiramente no mercado em 1910, e tinham nomes como "*One Man Pictures Orchestra*", "*Filmplayer*", "*Movieodeon*" e "*Pipe-Organ Orquestra*". Para além da música, estas máquinas forneciam uma série de efeitos sonoros. Os seus formatos iam desde um piano com um pequeno conjunto de várias percussões até máquinas complexas similares em tamanho a uma orquestra de vinte instrumentos, como é o caso do *Fotoplayer*.

[3] O presente artigo, o *patch* Pd, vídeos demonstrativos e a gravação áudio do concerto, podem ser descarregados no seguinte endereço electrónico: http://pedropatricio2008.googlepages.com/doutoramento

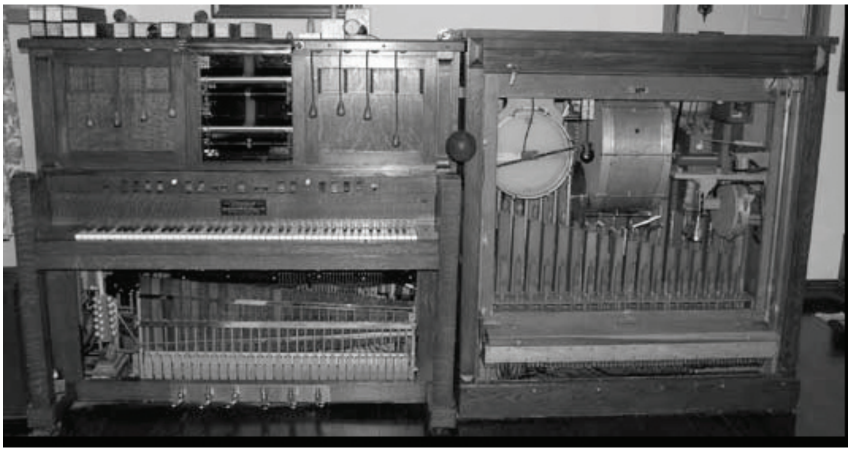


Figura 1. *Fotoplayer*

O *Fotoplayer* foi construído em 1926, em *Van Nuys* na Califórnia. É uma combinação de piano, órgão de tubos, percussão e vários efeitos sonoros desenhados para narrar a acção dos filmes mudos. Pedais, alavancas, interruptores, botões e cordas para puxar eram usados para accionar um xilofone, percutir um som, tocar um sino, criar o som de um relâmpago ou o chilrear de um pássaro. Quando os filmes sonoros começaram a impor-se nos anos 20, o *Fotoplayer,* foi-se tornando progressivamente num recurso obsoleto.

1.2. O papel da música e do som no cinema mudo

Nos primórdios do cinema mudo, o uso da música não derivava de qualquer necessidade artística ou psicológica, mas da necessidade de atenuar as péssimas condições acústicas das salas de projecção e substituir o barulho dos projectores por algo mais aprazível. Outro papel atribuído à música, era o de criar um ambiente sonoro propício e inspirador para os actores durante as filmagens. Muitas vezes usavam-se gravações em vez de música ao vivo, mas em ambos os casos a música servia sempre o mesmo propósito, o de inspirar os actores. Era também atribuído à música uma função ilustrativa, na tentativa de acrescentar emoções às imagens.

2. O Instrumento Musical Digital (IMD)

Na qualidade de compositor, decidimos dentro do paradigma do cinema mudo, da *performance* e da composição em tempo real, construir um instrumento musical digital que permitisse a sua utilização em concerto, que se apresentasse versátil, expressivo, que fosse de fácil manuseamento e que oferecesse possibilidades de se expandir e desenvolver no futuro. A criação deste tipo de instrumentos encontra as suas raízes nos trabalhos de Hans-Christoph Steiner (Reware prototypes in Untethered at Eyebeam, 2008) e Yann Seznec (Wii Loop Machine 2.0, 2008). Dentro deste panorama, escolhemos como dispositivo o iPod touch, cuja utilização como instrumento musical digital, pode ser observada em projectos como o *MoPho - the Mobile Phone Orchestra*, liderado por Ge Wang (CCRMA - Center for Computer Research in Music and Acoustics, Standford University)[4]

[4] Para um visionamento da peça musical *Gedrone,* ver http://www.youtube. com/watch?v=DhZ9g5U81io

Assim, para a construção do nosso instrumento recorremos ao seguinte triângulo tecnológico:

- o dispositivo: iPod Touch (versão 2.2);
- a aplicação: OSCRemote (versão1.2);
- o software: Pure Data (Pd).

2.1. O dispositivo: iPod Touch (versão 2.2)

O iPod touch foi projectado e introduzido no mercado em Setembro de 2007, pela Apple Inc. É um dispositivo multimédia portátil, um assistente digital pessoal e uma plataforma Wi-Fi móvel, que adiciona ao interface gráfico do utilizador o conceito do multi-toque *(touch screen)*.

Figura 2. iPod touch

2.2. A aplicação: OSCRemote (versão1.2)

OSCRemote é uma aplicação *Open Source* para o iPod touch e iPhone criada por Leo van der Veen (artista holandês ligado à Arte Multimédia Interactiva). É um controlador intuitivo, que ao usar o protocolo de comunicação *OpenSound Control*, transforma o iPod touch num controlador remoto, podendo ser utilizado por qualquer software (Max/MSP ou Pure Data), dentro de um sistema de rede WiFi. As principais características do OSCRemote são:

- Permite criar controladores dentro do editor.
- Permite salvar os ficheiros directamente no iPod touch.
- Confere a possibilidade de fazer *upload* de ficheiros de controlo.

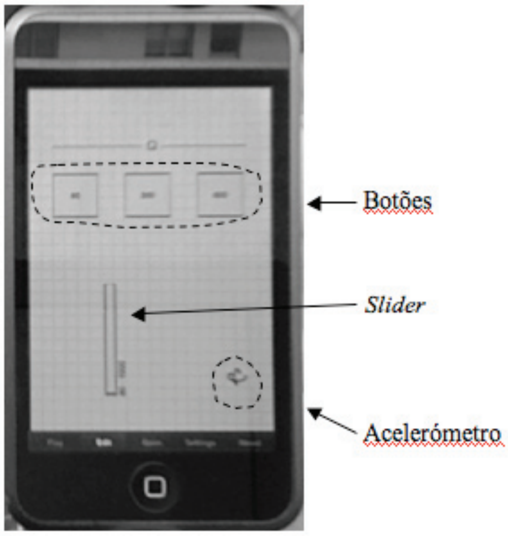


Figura 3. Interface gráfico do OSCRemote (modo de edição)

Dentro da lista de controladores que a aplicação possui foram usados os seguintes:

- Botão: envia o valor 1 (um) quando pressionado e o valor 0 (zero) quando libertado.
- *Slider*: envia valores continuadamente; pode-se estabelecer o valor mínimo e o valor máximo.
- Acelerómetro: envia a velocidade do movimento e a orientação espacial do dispositivo nas coordenadas x,y. O símbolo do Acelerómetro só é visível no modo de edição.

2.2.1. Comunicação entre o computador e o OSCRemote

- Para emitir os dados a partir da aplicação OSCRemote para o computador temos de saber qual é o endereço IP e a porta pela qual os dados irão ser recebidos (no caso do Pd utilizou-se a porta de entrada 1111).
- Para os computadores Macintosh, o IP pode ser encontrado na placa da rede das preferências do sistema OSX.
- O IP e a porta devem ser inscritos na configuração do OSCRemote. Por exemplo: IP: 192.168.1.101 e PortOut: 1111

2.3. O software: Pure Data (Pd)

O Pd (abreviatura de Pure Data) é uma linguagem de programação gráfica para áudio, vídeo e processamento gráfico em tempo real semelhante ao Max/MSP/Jitter. É o terceiro grande ramo da família das linguagens de programação conhecidas como Max (Max/FTS, ISPW Max, Max/MSP, jMax, etc.), originalmente desenvolvido por Miller Puckette (1987), e pelo Ircam. É um programa *Open Source* compatível com todos as plataformas e sistemas operativos. O núcleo do Pd escrito e mantido por Miller Puckette (1997), inclui também o trabalho de muitos colaboradores fazendo dele uma plataforma multimédia de cariz comunitário. Podem-se encontrar informações sobre estas colaborações em http://puredata.info/dev

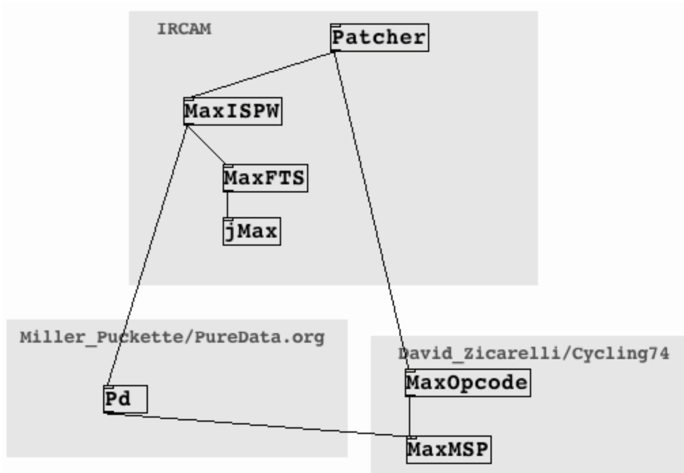


Figura 4. Árvore “genealógica” do Pure Data

3. Programação

3.1. O Interface

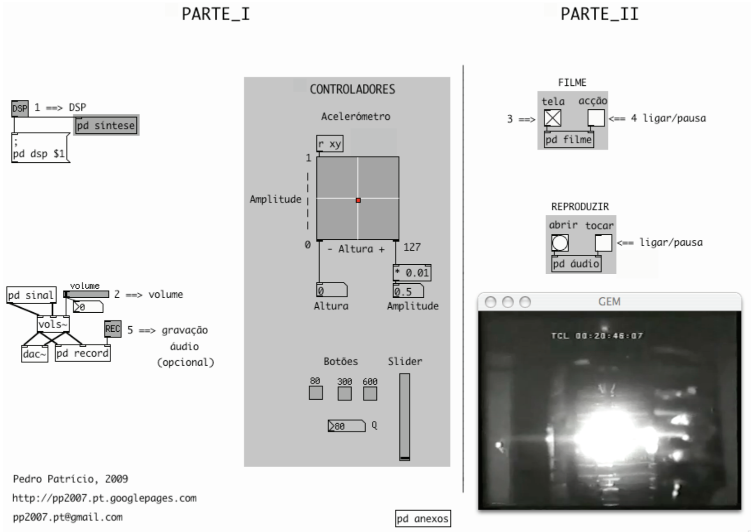


Figura 5. Interface

O Interface é constituído por duas partes. A primeira parte, considerada nuclear, é dedicada ao áudio e aos controladores. Fazem parte dela os seguintes *subpatches*:

- **pd síntese**: é um gerador de som constituído por ruído branco filtrado por um objecto chamado vcf~ *(voltage – controlled bandpass filter)*. O sinal áudio que sai do primeiro *outlet* do *subpatch pd OSCRemote* determina a frequência central, enquanto que o declive do filtro é determinado pelos valores do factor Q (razão entre a frequência central e a largura de banda), emitidos pelo *slider* e pelos botões.

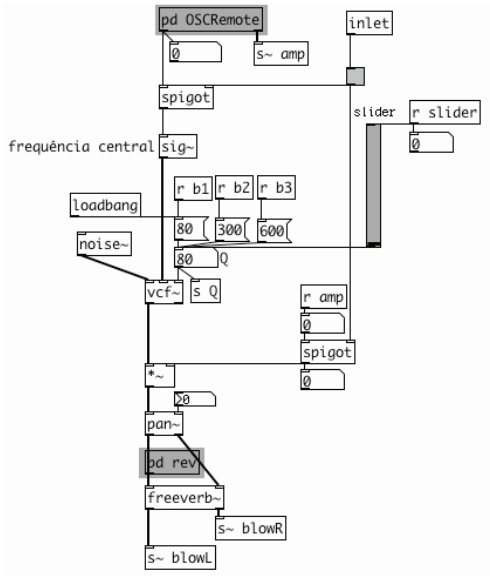


Figura 6. pd síntese *(subpatch)*

- **pd OSCRemote**: este *subpatch* é reservado à recepção e mapeamento dos dados emitidos pelos controladores da aplicação. Os valores emitidos pelo acelerómetro são normalizados de maneira a obter valores compreendidos entre zero e um para as amplitudes do som (coordenada x), e valores entre zero e 127, para as alturas do som (coordenada y).

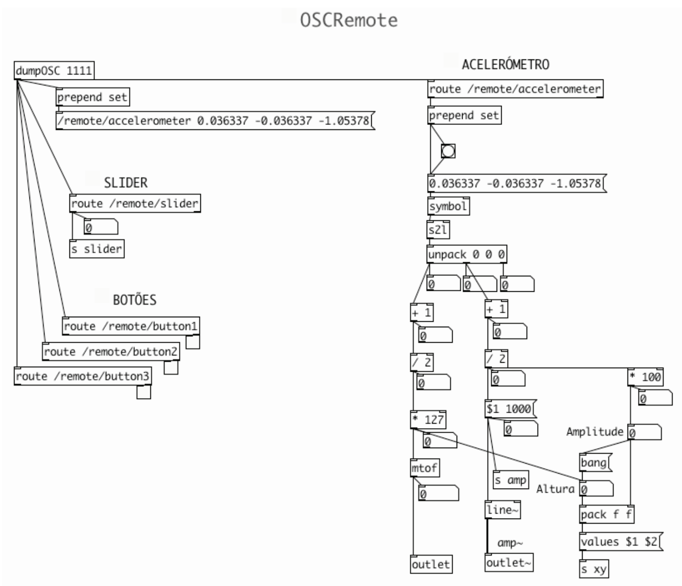


Figura 7. pd OSCRemote (mapeamento)

Podem-se visionar os valores do factor Q, a calibração do dispositivo e a reacção dos movimentos do utilizador através dos objectos chamados: *Graphical Users Interfaces* (GUI). Referimo-nos concretamente, aos objectos [grid], [toggle], [Vslider] e [nbx].

Tabela 1. Relaciona os controladores OSCRemote, os objectos GUI, e a função específica de cada um deles

Controladores OSCRemote	GUI	Função
Acelerómetro	[grid]	Permite visualizar a calibração do dispositivo e os valores centrais da amplitude (0.5) e das alturas (65).
Botões	[toggle]	Permite visualizar a activação dos valores pré-determinados do factor Q
Slider	[Vslider]	Permite visualizar e manipular o factor Q
Botões + Slider	[nbx]	Permite visualizar o valor Q

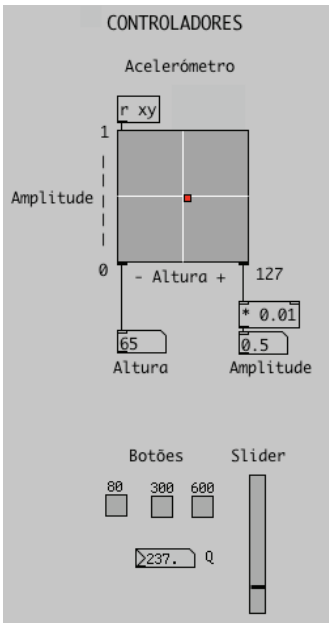


Figura 8. Controladores remotos

A segunda parte do Interface é destinada ao vídeo (visionamento do filme), à reprodução do áudio gravado e à sincronização vídeo/áudio. Seguindo a ordem numérica das etapas apresentadas no interface, ao accionar o interruptor designado de "tela" (etapa número 3), cria-se uma janela que permite o visionamento do filme seleccionado. O interruptor designado de "acção" (etapa número 4), funciona como os botões *Play* e *Pause* de um leitor de vídeo. Caso se opte por gravar o som (etapa número 5), poderemos posteriormente, reproduzi-lo sincronizado com o ficheiro de vídeo seleccionado. Basta accionar o interruptor "abrir" (etapa número 6), seguido do interruptor "tocar" (etapa número 7). A resolução do vídeo pode ser adaptada ao critério de cada utilizador e à resolução pré-definida do ficheiro de vídeo seleccionado. Para isso, é necessário enviar uma mensagem para o objecto [gemwin], do género: [dimen 640 480(

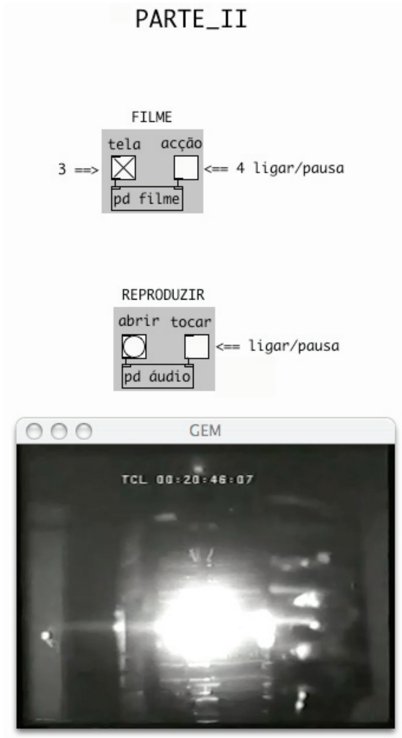


Figura 9. Parte II do interface

4. Manuseamento do Dispositivo

No modo de execução, com os valores enviados pelo acelerómetro obtiveram-se movimentos de rotação com ângulos compreendidos entre 0º e -180º para a coordenada x (amplitudes), e movimentos de rotação com ângulos compreendidos entre -90º e 90º para a coordenada y (alturas).

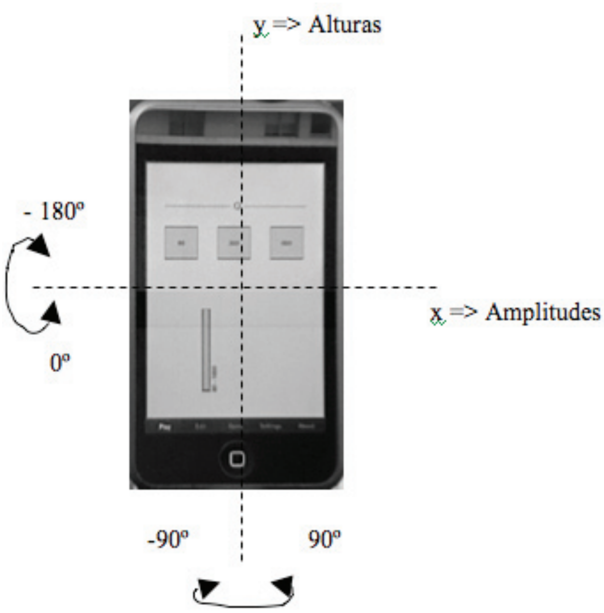


Figura 10. Interface gráfico do OSCRemote (modo de execução)

Para melhor visualização elaboraram-se as seguintes sequências de imagens:

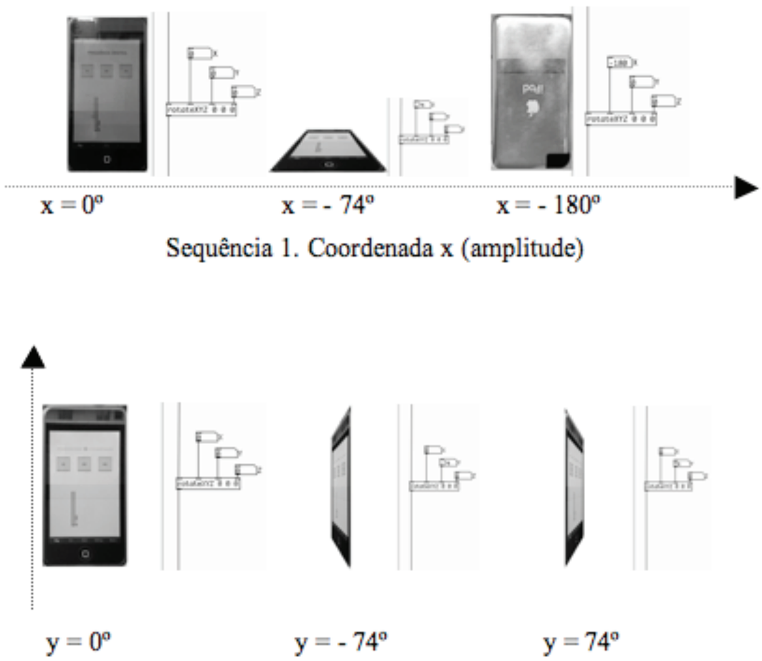


Figura 11. Ângulos de rotação nas coordenadas x,y

De acordo com os valores escalonados recorreu-se a movimentos de flexão, extensão e rotação dos pulsos para manusear o dispositivo. A relação entre os controladores, os movimentos dos pulsos, os gestos dos dedos e a respectiva função encontra-se esquematizada na tabela 2.

Tabela 2. Sumário dos controladores OSCRemote, movimentos dos pulsos, gestos e respectiva função

Controladores	Movimento/Gesto	Função
Acelerómetro	Flexão e extensão dos pulsos	Aumenta ou diminui a amplitude
	Rotação interna e externa dos pulsos	Aumenta ou diminui a frequência central do objecto [vcf~]
Botões	Pressionar	Selecciona o factor Q (80, 300, 600)
Slider	Mover cursor	Aumenta ou diminui o factor Q (min: 80, máx: 1000)

O Acelerómetro, ao permitir que a velocidade de movimento, a altura e a amplitude do som gerado respondam instantaneamente a cada movimento da mão do *performer*, faz do instrumento musical digital utilizado, um dispositivo inerentemente reactivo e expressivo.

5. Composição Musical – opções estéticas

Utilizámos como máquina viajante a metáfora sonora da voz do vento construído a partir de ruído branco filtrado. Apesar de não se retratar musicalmente nenhum ambiente acústico em particular, o ouvinte poderá relacionar a música com o som específico de um elemento natural do meio ambiente, aproximando as opções estéticas da composição de alguns aspectos relacionados com os princípios da *Soundscape*

Composition e da Ecocomposição (Schafer, 1977; Truax, 1996; Keller, 1999; Burtner, 2005).

5.1. O Guião

No guião, utilizámos algumas referências ou separadores para sabermos que em determinados pontos do vídeo, musicalmente algo teria que acontecer. Os separadores que considerámos mais importantes foram: (a) O mergulho da ponte. (b) O primeiro barco. (c) As pessoas no mercado. A estes separadores fizemos corresponder alguns gestos sonoros: (a) Som linear + glissando ascendente até à ponte[5]. (b) Diminuição da amplitude do som[5]. (c) Pequenos apontamentos rítmico-melódicos quando surgem as pessoas no mercado originados a partir de movimentos rápidos e bruscos de rotação interna e externa dos pulsos[6]. (d) F*ade out* final[6].

6. Perspectivas

O IMD proposto e descrito neste artigo, revelou ser funcional (1) por ser de tamanho reduzido, (2) utiliza comunicação sem fios, (3) requer gestos finos de motricidade (4) é discreto em concertos, (5) é de fácil manuseamento e permite obter resultados musicais expressivos. Estes dispositivos multimédia portáteis, vulgares e de "uso comum" (o Laptop e o iPod touch), em contextos artísticos, apresentam a vantagem de não ser necessário efectuar grandes investimentos na compra ou na construção de raiz de protótipos de IMD por empresas específicas.

No futuro poder-se-á (1) melhorar e aumentar as possibilidades sonoras, introduzindo osciladores, panorâmicas, leitura variável de *samples* e utilização de outras técnicas de síntese som, por exemplo: síntese aditiva, modulação de frequência, modulação em anel, processamento de efeitos, etc, (2) utilizar outros controladores disponíveis na aplicação, por exemplo: *XY Pad*, para relacionar e manipular simultaneamente 2 parâmetros de controle, sonoros ou musicais; e o *Switch*, para despoletar eventos sonoros, ligar e desligar automatizações, etc, (3) realizar um vídeo onde se possa observar a utilização do instrumento durante uma *performance* ao vivo.

7. Agradecimentos

Agradecemos a Luís Gustavo Martins e a António de Sousa Dias as sugestões e comentários críticos.

8. Referências

Burtner, M., (2005) "Shamanic and Ecoacoustic Technologies for Multimedia Composition and Performance", *Organized Sound*. Volume 31, Cambridge Press, York, England.

Cornelis, O., Demey, M., Leman, M., (2008) "EME: a Wireless Music Controller for Real-time Music Interaction", Artech, 4th International Conference on Digital Arts, 7- 8 November, Portuguese Catholic University, Porto.

[5] Ver demonstração 5 em http://pedropatricio2008.googlepages.com/doutoramento

[6] Ver demonstração 6 em http://pedropatricio2008.googlepages.com/doutoramento

Keller, D. (2000) “Compositional processes from an ecological perspective”, *Leonardo Music Journal,* 10, 55-60.

Keller, D., & B. Truax (1998) “Ecologically-based granular synthesis. *Proceedings of the International Computer Music Conference*”, Ann Arbor, MI: ICMA.

Puckette, M., (2002) "Max at seventeen", Computer Music Journal 26(4): pp. 31-43.

Puckette, M., (1996) "Pure Data: another integrated computer music environment" Proceedings, Second Intercollege Computer Music Concerts, Tachikawa, Japan, pp. 37-41.

Ruoff, J. K., (1992) “Conventions of sound in documentary”, In R. Altman (Ed.), *Sound Theory Sound Practice* (pp. 217–234). New York: Routledge.

Schafer, R. M. (1977) “*The Tuning of the World”*, New York: Knopf; reprinted as *The Soundscape: Our Sonic Environment and The Tuning of the World*. Rochester, Vt: Destiny Books, 1994.

Truax, B. (1996) “Soundscape, acoustic communication & environmental sound composition”, *Contemporary Music Review* 15(1): 49-65.

Wang, G., Essl, G., Penttinen, H., (2008) “MoPhO: Do Mobile Phones Dreams of Electric Orchestras?”, In Proceedings of the International Computer Music Conference. Belfast, 2008.

Wang, G., Cook, R. P., (2004) “On-the-fly Programming: using Code as an Expressive Musical Instrument”, In Proceedings of the International Conference on New Interfaces for Musical expression (NIME).

http://www.nr37.nl/OSCRemote/ (última consulta em 2009. 05. 9).

http://puredata.info/ (última consulta em 2009. 05. 21).

http://cvc.instituto camoes.pt/cinema/documentarismo/doc005.html (última consulta em 2009. 05. 11).

http://www.scoretrack.net/musicadecinema.html (última consulta em 2009. 05. 15).

http://www.apple.com/ipodtouch/ (última consulta em 2009. 05. 19).

http://en.wikipedia.org/wiki/Camille_Saint-Saëns (última consulta em 2009. 05. 22).

http://en.wikipedia.org/wiki/The_Assassination_of_the_Duke_of_Guise (última consulta em 2009. 05. 29).

http://filmsdefrance.com/FDF_Assassinat_du_duc_du_Guise_rev.html (última consulta em 2009. 05. 29).

http://www.victorianreproductionlighting.com/antique-photoplayer.html (última consulta em 2009. 06. 01).

http://www.theamazingrolo.net/ (última consulta em 2009. 06. 01)

http://at.or.at/hans/works.html (última consulta em 2009. 06. 01)

O Mapa de Hénon como Gerador de Repositórios Composicionais

Liduino José Pitombeira de Oliveira[1], Hildegard Paulino Barbosa[2]

[1]PPGM/Departamento de Música/MUS**3** – Universidade Federal da Paraíba (UFPB)

[2]Departamento de Informática/MUS**3** – Universidade Federal da Paraíba (UFPB)
Cidade Universitária – João Pessoa – PB – 58059-900 – Brazil

pitombeira@yahoo.com, hildegardpaulino@gmail.com

Resumo. *Este artigo trata da utilização do sistema caótico bidimensional denominado Mapa de Hénon como gerador de repositórios composicionais. A criação de um aplicativo em Java, que automatizou o modelo matemático deste sistema, possibilitou a identificação de padrões de notas, classes-de-notas (ordenadas e desordenadas) e formas primas e, a partir daí, o planejamento de uma obra para oboé, fagote e piano.*

Abstract. *This paper describes the application of the chaotic system called Hénon map as the generator of a repository to be used in compositional design. The elaboration of a computer program in Java, which automated the mathematical model of this system, made it possible to identify patterns in terms of pitch, pitch-class (ordered and unordered) and prime forms, and, thus, to plan a piece for oboe, bassoon, and piano.*

1. Considerações Gerais

Este artigo trata da utilização do sistema não-linear denominado Mapa de Hénon como gerador de repositórios que auxiliaram no planejamento composicional de uma obra para oboé, fagote e piano. Um aplicativo Java, criado durante a pesquisa, possibilitou a realização de experimentos com vistas à identificação de padrões e à geração de um banco de parâmetros para fins composicionais. Este banco de parâmetros se assemelha aos *chart systems* que John Cage utilizou como repositórios de parâmetros (sonoridade, duração e dinâmica) durante a composição do "Concerto for Prepared Piano" [Pritchet 1996]. A diferença é que, enquanto Cage decidiu a ordenação dos parâmetros com o auxílio do *I Ching,* a ordenação realizada aqui se deu por uma hierarquização baseada na estatística dos padrões encontrados pelo aplicativo, bem como pela livre escolha de gestos da camada superficial literalmente fornecidos pelas equações. Depois de uma breve introdução sobre caos, faremos um detalhamento histórico da pesquisa e a descrição do aplicativo. Em seguida, abordaremos as fases de planejamento composicional que culminaram na criação da obra para oboé, fagote e piano intitulada "Hénon".

Para Bidlack [1992], caos "é o termo genérico usado para descrever a saída, sob certas condições, de sistemas dinâmicos não-lineares". Mandelbrot [1982] acrescenta que em um comportamento caótico, "nenhum ponto é visitado duas vezes em um tempo finito". Já Moon [2004] diz que sistemas caóticos têm seu comportamento sempre previsível e que a incerteza do estado atual de um sistema caótico cresce exponencialmente com o passar do tempo. Comportamentos não-lineares caóticos são encontrados, por exemplo em diodos, forças magnéticas e elétricas, elementos de capacitância, indução e resistência de circuitos e transistores [Moon 2004].

2. O Mapa de Hénon

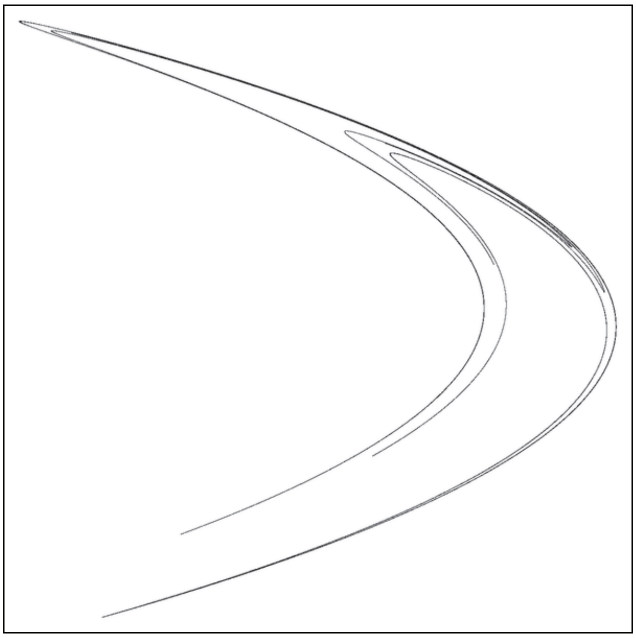


Figura 1. Mapa de Hénon

Nosso ponto de partida foi o exame de um sistema caótico bidimensional denominado Mapa de Hénon, o qual não é derivado de nenhum fenômeno natural em particular [Bidlack 1992]. Este sistema é expresso através das equações $x_{n+1} = y_{n+1}\text{-}Ax_n^2$ e $Y_{n+1} = Bx_n$, onde A e B são constantes positivas. Para valores de A=1,4 e B=0,3 o sistema atinge um estado caótico gerando o atrator mostrado na Figura 1. Por ser bidimensional, os resultados deste sistema podem ser associados a somente dois parâmetros musicais. No nosso caso, associamos estes resultados aos parâmetros altura e ritmo (ponto de ataque e duração). A altura pode ser tratada nos formatos de: [1] notas, onde os registros são considerados e apenas ajustados em suas oitavas, quando necessário, para a extensão instrumental; [2] classes-de-notas ordenadas ou desordenadas (o termo classe-de-notas se baseia na tradução, ainda não publicada, da obra *Introduction to Post-Tonal Theory* de Joseph Straus, realizada pelo Dr. Ricardo Bordini, da Universidade Federal da Bahia); ou ainda [3] formas primas. Para a estrutura rítmica, elaboramos uma tabela de equivalência (Tabela 1) entre os valores numéricos de saída e as figuras rítmicas correspondentes, devidamente quantizadas por aproximação (por exemplo, um resultado igual a 1260 foi associado à semicolcheia), para que pudessem ser convenientemente traduzidas por um aplicativo de notação musical. O uso de pausas se constituiu em uma abertura no sistema composicional, ou seja, foram utilizadas livremente pelo compositor.

Tabela 1. Equivalência entre valores numéricos e figuras rítmicas

Figura	Valor
Mínima	10080
Semínima pontuada	7560
Semínima	5040
Colcheia pontuada	3780
Semínima quialtera de 3	3360
Colcheia	2520
Colcheia quialtera de 5	2016
Semicolcheia pontuada	1890
Colcheia quialtera de 3	1680
Colcheia quialtera de 7	1440
Semicolcheia	1260
Semicolcheia quialtera de 5	1008
Semicolcheia quialtera de 7	720
Fusa	630

3. O Aplicativo Java

A Figura 2 mostra a interface gráfica do programa desenvolvido durante a pesquisa, o qual já manipula o Mapa Logístico, o Mapa de Hénon e o Conjunto de Mandelbrot. A rotina de uso consiste em escolher: [1] o tipo de fractal, que no nosso caso será o Mapa de Hénon; [2] os valores para A e B, dentro dos domínios mencionados anteriormente; [3] o número de interações; [4] o tipo de gráfico (cartesiano, atrator ou plotagem ao infinito, a qual não será tratada neste artigo); [5] o tipo de saída do arquivo MIDI (notas, classes-de-notas ou formas primas); e [6] o tipo de padrão a ser buscado (tricordes, tetracordes, pentacordes ou hexacordes). Ao se pressionar o botão COMEÇAR, o programa gera: [1] um gráfico dentro da própria interface (canto superior direito), o qual pode ser percorrido em toda sua extensão com o auxílio de duas setas de deslocamento posicionadas abaixo do mesmo ou salvo no formato PNG para análise posterior; [2] um arquivo MIDI tipo 1 (este formato foi escolhido pela portabilidade, isto é, pela facilidade de conversão dos dados em notação musical convencional, bem como pela possibilidade de expansão do aplicativo, quando da futura necessidade da associação de uma das dimensões ao parâmetro timbre); e [3] tabelas de padrões, no canto inferior esquerdo, as quais podem ser salvas no formato TXT (esses dados foram mostrados já convenientemente dispostos na partitura da Figura 3 e na Tabela 2). Estas tabelas juntamente com o arquivo MIDI, que pode ser manipulado em um programa de edição de partitura (FINALE, por exemplo) para melhor visualização, são os dados mais importantes durante o planejamento composicional. O algoritmo central que realiza o cálculo dos valores para o Mapa de Hénon é:

```
for (int i = 0; i < iteracoes; i++) {
        new_x = y+1.0-(a*(x*x));
        new_y = b*x;
        x = new_x;
        y = new_y;
        resultadox = x % modulo;
```

```
            resultadoy = y;

            Double numx = new Double(resultadox), numy = new
Double(resultadoy);
            if (!numx.isNaN() && !numy.isNaN() && !numx.isInfinite() &&
!numy.isInfinite()) {
               notasProv.inserirDado(resultadox);
               duracoesProv.inserirDado(resultadoy);
            }
            else {
               break;
            }
         }
```

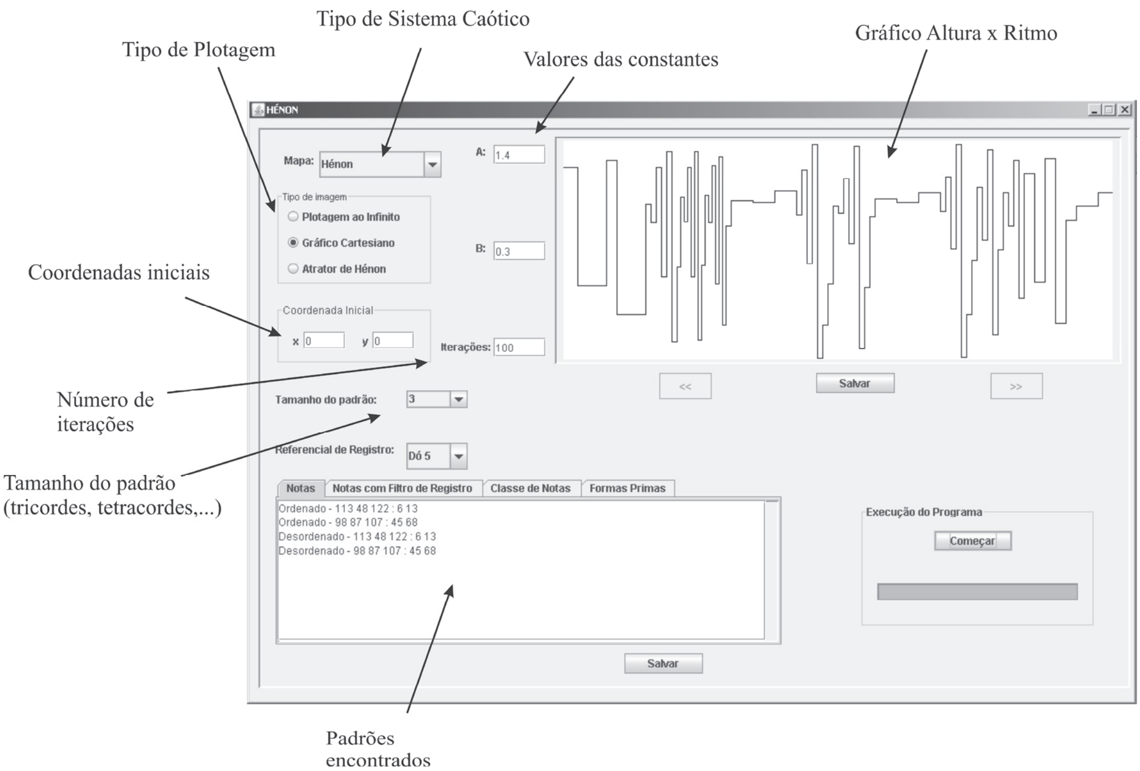


Figura 2. Interface do usuário para o Mapa de Hénon

Os dados iniciais que escolhemos foram: [1] tipo de fractal: Mapa de Hénon; [2] os valores de A e B ficaram fixos em 1,4 e 0,3, respectivamente (estado caótico); [3] 100 iterações; [4] a plotagem foi a cartesiana, onde o eixo das ordenadas é associado ao parâmetro altura e o eixo das abscissas ao parâmetro ritmo, que se compõe de ponto de ataque e duração; [5] o arquivo MIDI gerado contém classes-de-notas, isto é, sem consideração de registro; e [6] buscamos padrões tricordais. A partir dos dados gerados examinamos os arquivos MIDI e a tabela de padrões, os quais são identificados pelo aplicativo, buscando detectar recorrências de conjuntos de classes-de-notas desordenados, que pudessem se configurar como estruturas quantitativamente importantes no planejamento. Isto nos permitiu organizar repositórios onde se observaram tendências do sistema em produzir padrões específicos que permitiram pensar em um equilíbrio entre diferença e repetição. Na Figura 3, temos o resultado das cem iterações convertidas para notação musical, através do programa FINALE, e os padrões identificados com um rótulo (letras do alfabeto), indicando o conjunto de classe-de-notas e a forma prima deste conjunto entre parênteses. A Tabela 2 mostra os padrões de conjuntos de classes-de-notas desordenados. Estes conjuntos foram rotulados com base em suas formas primas. Utilizamos apenas os padrões quantitativamente significantes, isto é, com no mínimo quatro ocorrências. Só estes sete padrões (A, A', B, C, D, D', E), onde dois pares são intimamente relacionados pela forma prima (A/A' e D/D') receberam rótulos e foram utilizados como repositórios de alturas na obra.

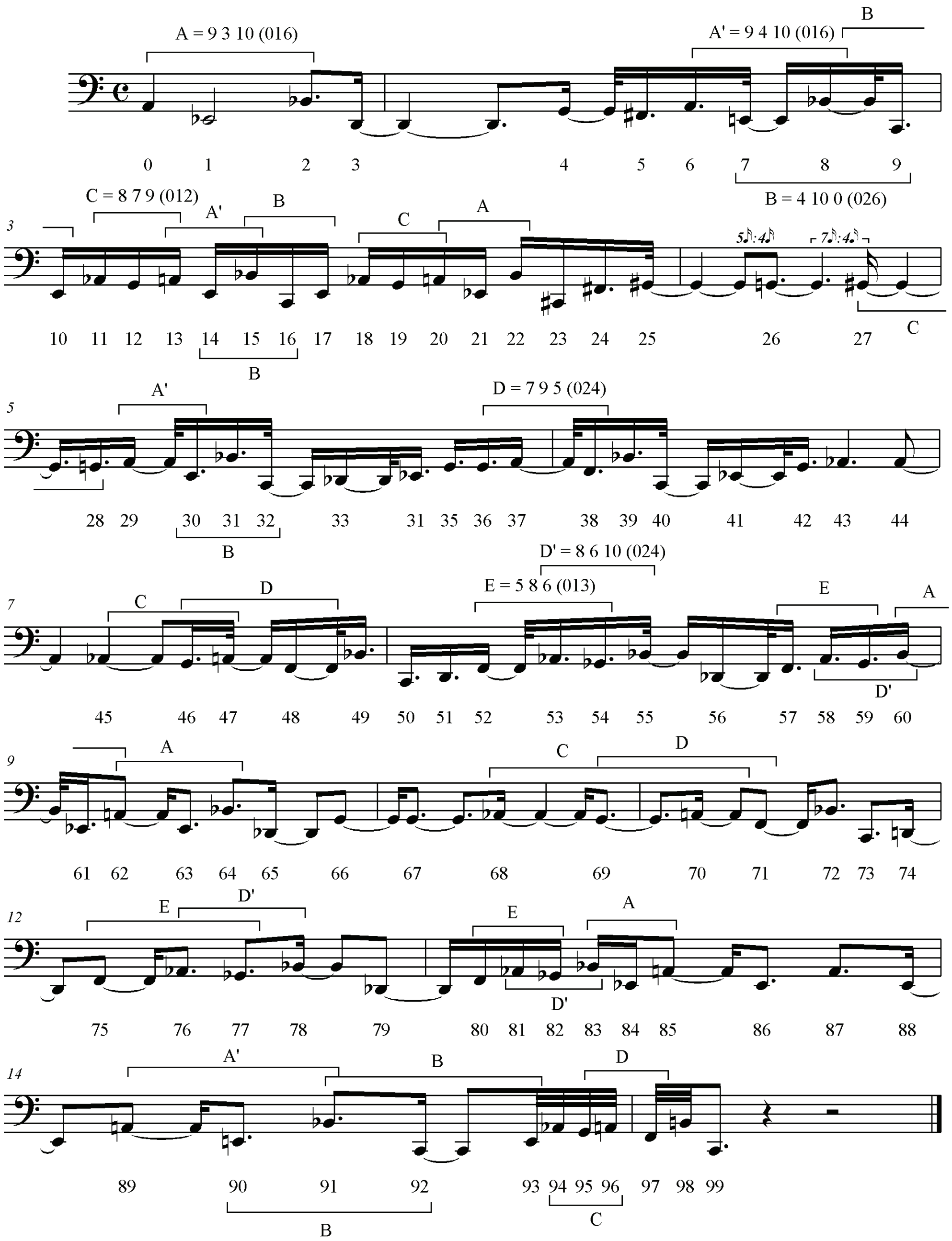


Figura 3. Resultado das iterações convertido para notação musical

Tabela 2. Padrões de conjuntos tricordais de classes-de-notas desordenados

Rótulo	Forma prima	Conjunto de Classes-de-notas	Ponto de ocorrência	Ocorrências
A	016	9 3 10	0 20 60 62 83	5
A’	016	9 4 10	6 13 29 89	4
B	026	4 10 0	7 8 14 15 30 90 91	7
		0 4 8	9 16 92	3
		4 8 7	10 17 93	3
C	012	8 7 9	11 18 27 45 68 94	6
		7 9 4	12 28	2
		3 10 1	21 63	2
		10 1 6	22 54 77	3
D	024	7 9 5	36 46 69 95	4
		9 5 10	37 47 70	3
		5 10 0	38 48 71	3
		10 0 2	49 72	2
		0 2 5	50 73	2
		2 5 8	51 74	2
E	013	5 8 6	52 57 75 80	4
D’	024	8 6 10	53 58 76 81	4
		10 1 5	55 78	2
		1 5 8	56 79	2
		6 10 3	59 82	2

4. Planejamento Composicional

Os padrões gerados pelo sistema caótico foram utilizados de duas maneiras: a) literalmente, na forma determinística com que foram produzidos; b) hierarquizados a partir de suas formas primas. Isto possibilitou a convivência simultânea entre dados em estado bruto (fornecidos pela equação) e dados tratados hierarquicamente. A transição entre esses dois estados, assim como o uso de pausas mencionado anteriormente, se constituiu em abertura no sistema composicional, sendo portanto de uso livre.

4.1. O parâmetro altura

Os conjuntos de classes-de-notas foram hierarquizados tomando como critério o número de ocorrências. Desta forma, o conjunto {4,10,0}, que aparece sete vezes, tem o maior nível hierárquico, seguido dos conjuntos {8,7,9}, {9,3,10}, {9,4,10}, {7,9,5}, {8,6,10} e {5,8,6}. Os quatro primeiros conjuntos foram utilizados como pilares estruturais de acordo com o diagrama mostrado na Figura 4, onde podemos observar como os conjuntos {4,10,0}, {8,7,9}, {9,3,10} e {9,4,10} têm a função de demarcar formalmente as seções da obra. Na seção A, as classes-de-notas do conjunto {4,10,0} são salientes. Este conjunto também emoldura a obra integralmente e marca o retorno da seção A variada, onde aparecem as classes-de-nota 3 e 4 para formar concomitantemente os conjuntos {9,3,10} e {9,4,10}, ambos com forma prima 016. A seção central B foi construída a partir do conjunto {8,7,9}, que sugere gestos abundantes em intervalos de segunda menor.

Tanto os conjuntos literais produzidos pelo sistema caótico, como suas formas primas, expressas sob a forma de outras configurações de classes-de-notas, isto é, transpostas, invertidas, desordenadas etc, foram utilizados na obra. Também serão utilizadas citações literais do arquivo MIDI no formato sequencial com que foram geradas pelo Mapa de Hénon (Figura 3).

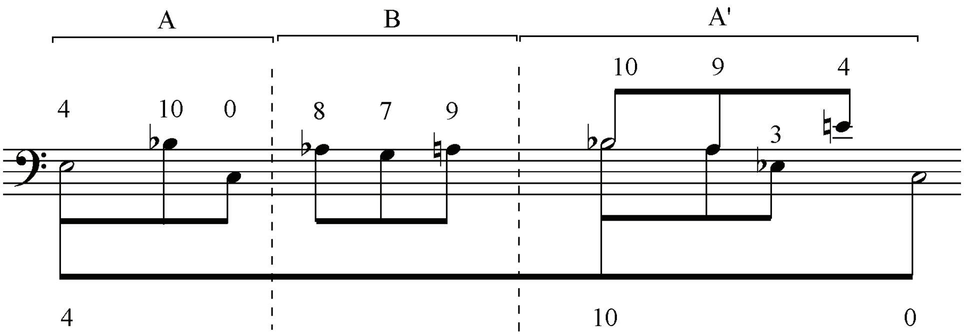


Figura 4. Planejamento estrutural da obra em função dos conjuntos

4.2. O parâmetro ritmo

Assim como cinco sonoridades primas (A,B,C,D,E) foram detectadas nos resultados das iterações do Mapa de Hénon e foram classificadas em sete padrões de classes-de-notas, cinco padrões foram detectados nos gestos rítmicos, a partir de uma moldura métrica quaternária. Evidentemente, um olhar da estrutura rítmica, a partir de outros ângulos, poderia identificar padrões de outra natureza. Por exemplo, na Figura 3, encontramos uma série contínua de semicolcheias pontuadas entre os eventos 28—42 e 46—61, bem como uma série contínua de colcheias pontuadas entre os eventos 62—79. No entanto, os padrões aqui utilizados, identificados na Figura 5, foram observados (como uma escolha na fase de planejamento composicional) a partir de uma pulsação constante de semínimas em compasso 4/4. Tais padrões foram utilizados em abundância na obra, tanto no formato literal como alterado. Os procedimentos de alteração consistiram em: a) dissociar esses padrões rítmicos das alturas fornecidas pelo sistema caótico; b) utilizar esses padrões em sequências temporais diferentes das originais.

4.3. Gestos iniciais da obra

A Figura 6 mostra a página inical de “Hénon”, para oboé, fagote e piano. A obra se inicia utilizando um fragmento do primeiro gesto rítmico (Figura 5) adaptado intervalicamente ao conjunto 026, de maior nível hierárquico, o qual se inicia na classe-de-notas 4 (Mi), definida como estrutural no planejamento composicional (Figura 4). Um agregado de doze sons construído totalmente com o conjunto 026 é distribuído nos três primeiros compassos (4,10,0,3,7,9,2,6,8,1,5,11) ao mesmo tempo em que a classe-de-notas estrutural 4 é prolongada até o início do segundo gesto (compasso 4). No segundo compasso, o quinto gesto rítmico é também adaptado intervalicamente ao conjunto 026. Os compassos 1-3 integram o que poderíamos designar de frase 1. A segunda frase também salienta a classe-de-nota estrutural 4, a qual aparece em todos os instrumentos. Em seguida, o gesto inicial gerado pelo sistema de equações (Figura 3) é transposto para iniciar na classe-de-nota 4, primeiramente no oboé e em seguida no fagote, o qual prossegue mostrando duas configurações do conjunto 026 no compasso 7. No compasso 8, uma citação literal do gesto gerado pelas equações é mostrada inicialmente em todos os instrumentos sendo finalizada somente pelo piano.

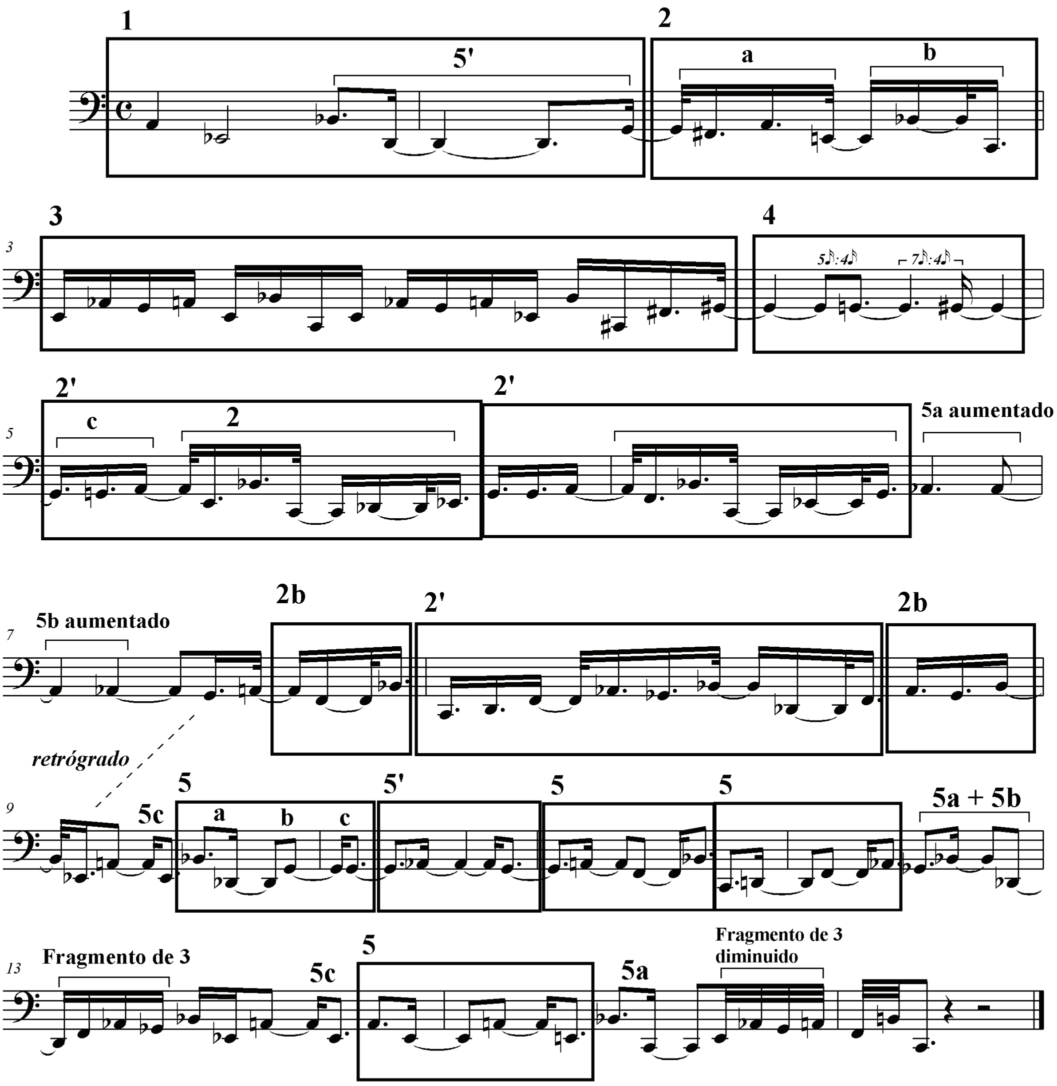


Figura 5. Padrões rítmicos

5. Conclusão

A obra "Hénon", para oboé, fagote e piano, criada a partir de um sistema caótico denominado Mapa de Hénon, cujos valores foram filtrados e hierarquizados em uma fase de planejamento composicional, demonstra o emprego concomitante de princípios matemáticos, aplicativos computacionais e sistemas composicionais. A convivência de duas disciplinas científicas (matemática e informática) com a música, em torno de um objetivo comum, no caso, a criação de uma obra de arte musical, nos reportam às originais conexões entre a música e a matemática, explícitas até o século XVIII, no âmbito do conjunto de disciplinas denominado quadrivium (aritmética, geometria, astronomia e música). Tais conexões foram aparentemente um pouco atenuadas pela incursão da música nas disciplinas do trivium (gramática, retórica e lógica), através da extensiva aplicação da retórica na composição e análise de obras, a partir do período barroco. No entanto, aplicações da matemática na música, vêm retornando cada vez com maior intensidade, desde a década de 1960, no campo analítico (vejam-se, por exemplo, os diversos artigos publicados no *Music Theory Spectrum*, *Journal of Music Theory* e

Perspectives of New Music, que claramente utilizam a matemática como ferramenta auxiliar na análise musical) e no campo composicional, não só através do uso de procedimentos algorítmicos, mas também na aplicação de estruturas e conceitos matemáticos de forma arquetípica ou metafórica, como, por exemplo, na obra "Dust", escrita em 2003 pelo compositor espanhol Francisco Lara, que recorre à geometria da Poeira de Cantor como um importante delineador estrutural. Esta pesquisa se insere nesta realidade.

Figura 6. Página inicial de Hénon

6. Referências

BIDLACK, Rick. Chaotic Systems as Simple (but Complex) Compositional Algorithms. *Computer Music Journal*, v. 16, n. 03, p. 33 – 47, Fall 1992.

MANDELBROT, Benoít B. *The Fractal Geometry of Nature*. New York: W. H.Freeman, 1982.

MOON, Francis C. *Chaotic Vibrations An Introduction for Applied Scientists and Engineers*. New York: John Wiley & Sons, 2004.

PRITCHET, James. *The Music of John Cage.* New York: Cambridge University Press, 1996.

12°

Simpósio Brasileiro de
Computação Musical

12th Brazilian Symposium on Computer Music

POSTERS

Self-Regulatory Feedback Systems as Sound Installations

Paula Matthusen

Florida International University – School of Music – Miami, FL 33199 – USA

matthuse@fiu.edu

Abstract. *This paper examines the development of two recent sound installations,* Filling Vessels *and* circadia*, and their method for exciting and controlling feedback through adopting techniques associated with a-life. The evolution of these installations is traced, while examining the decentralized performance encouraged within them.*

1. Introduction

Filing Vessels and *circadia* are two related sound installations developed in order to examine ways in which people interact with an acoustical space and subtly impact the sounds within it. Of particular interest and influence was Alvin Lucier's work as detailed in *Reflections* (1995), and especially his sound installation *Empty Vessels* (1997) in its striking manner of generating feedback. In this installation, Lucier inserted microphones into jars, and vases that were distributed throughout the space. The input on the microphones was then amplified such that they were on the cusp generating feedback. This extremely delicate balance was navigated by the visitors to the sound installation, as their movement through the space affected resonances within it.

Empty Vessels provided an initial impetus to experiment with feedback in a similar manner; that is, using feedback as a means of exploring the acoustical qualities of a particular space and the impact visitors have on it through their presence and movement. This interest aligned with the goal of making a feedback-based sound installation self-regulatory, cajoling itself towards producing various tones, and curtailing itself from spinning wildly out of control. Of particular interest was how a sound installation might acquire a unique sonic identity based on the self-regulatory mechanism employed. This paper traces the evolution and gradual merging of *Filling Vessels* and *circadia,* discussing the manner in which both systems regulate themselves.

2. *Filling Vessels*

The development of *Filling Vessels* began during the Fall of 2005. The title chosen pays homage to Lucier's pioneering work with feedback and the pivotal role *Empty Vessels* had in inspiring this work. *Filling Vessels* began by experimenting with generating feedback by amplifying microphones placed in jars until they were at the threshold of generating feedback. The sensitivity of the microphones to the acoustical properties of the space and one's location within it were immediately evident. Feedback would at times emerge and then cease if one walked to a different part of the room upon excitation. One could also easily hear how specific pitches and tones became excited simply talking, clapping, and singing. At times, the feedback could begin, and then crescendo to nearly unsafe volume levels. Utilizing "adaptive games," as discussed by Eduardo Miranda (2003), served as a model for reigning in the feedback.

Adaptive games utilize agent-based behavior in systems drawing on tenets of a-life that do not specifically involve cellular automata or genetic algorithms, but instead employ algorithms related to the interaction of multiple agents with a bottom-up emphasis, such as flocking and swarming. In particular, I was interested in what I call *decentralized performance*—that is, the specific actions and performative responsibilities of agents within systems that involve the aforementioned techniques.

The goal was to develop a system that could keep track of itself in a bottom-up manner. Instead of having a global control of amplification, each individual feedback-generating vessel is treated as an independent agent. Four vessels are located at the center of the room, with a microphone inserted in each jar. The signal feedback generated is routed to four different speakers located at the edges of the installation space. Each vessel is assigned its own speaker. The feedback signal passes through a Max/MSP patch tracking the overall volume and pitches produced by each of the four vessels. Each vessel tracks its own activity in relation to the others, making sure that each vessel has a chance to generate its own, unique feedback. Large cascading waves of feedback emerge with tones arising from one vessel while another decrescendos.

Most crucial in this particular feedback scenario is determining the rate at which the amplification is gradually decreased and increased. This calibration changes wildly from space to space, depending on the shape and resonance of the room and the number and location of people within it. The overall tones produced depend on the resonance of the jars, the room, the equipment mediating the feedback, and a shifting filter system that changes every time a pre-specified volume level is exceeded. Another element introduced to the system was to have it constantly recording and cataloging recorded tones based on register. These samples are reheard occasionally as shifting, discrete rhythms that are selected based on the frequencies present in the room. This lends the space a memory of the sounds that have occurred within it, and provides small perforations of the larger waves of sound from the four vessels. *Filling Vessels* later acquired a video element in collaboration with Tom O'Doherty, consisting of images of light reflected in the performance space. After working with the large waves of sound in *Filling Vessels,* I was interested in exploring smaller, waves of sound. This interest fueled the development of *circadia*.

3. *circadia*

circadia began as an outgrowth of *Filling Vessels,* utilizing a similar manner for producing feedback. The focal interest shifted to creating small waves of sound by using eight glass jars of various sizes as the feedback-producing vessels that doubled as the resonating bodies for speakers mounted on top of the jars. The speakers, in this case, are actually the lids to the jars, enclosing a small condenser microphone and LED inside the vessel. The microphones are amplified such that feedback emerges out of each jar, and is routed out its own speaker-lid. Thus, one can listen to the unique feedback each vessel generates.

Each vessel in *circadia* is treated as a separate agent as well. In contrast to *Filling Vessels*, these vessels work towards synchronization. The amplification of each microphone is controlled through a Max/MSP patch specifying different envelopes that determine the gradual increase and decrease of the amplification. A series of 12 different envelopes are possible, and each one can be triggered at different rates. A vessel is randomly assigned one envelope and pulsation rate at the beginning of the synchronization process. The vessels work towards synchronization by finding a common rate of pulsation; that is, how often the envelopes are triggered, and then by converging on a common envelope. No single vessel acts as a leader. Instead, each one takes on the pulsation rate of a neighboring vessel depending on whether or not it is more active than itself, resulting in shifting pulsations of subtle feedback until finally coalescing into one shared pulse. The LEDs within each jar are controlled by the amplitude of the signal, reflecting these pulsations with light. The overall look resembles fireflies flickering in the night, referencing the childhood experience of watching the visual dance of these insects.

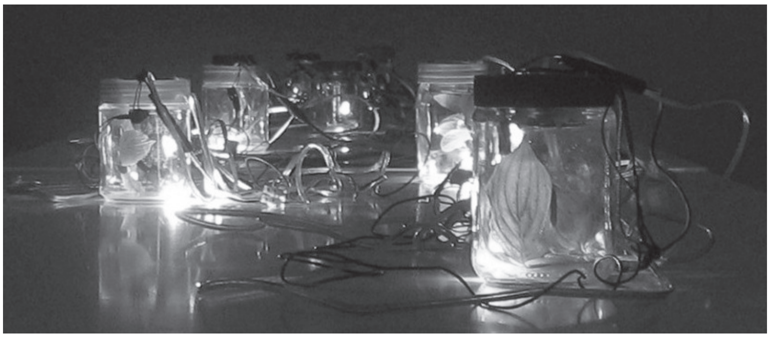


Figure 3. Image of *circadia.*

circadia, like *Filling Vessels*, is impacted by the acoustic properties of the space as well as the audience's interaction with it. Frequently, visitors to the installation would talk and sing into the vessels. By exciting resonances within the jars in this manner, the tones of the feedback would often change based on what frequencies were excited. Intriguingly, by working with the synchronization algorithm, the vessels would often excite different frequencies in one another. This was a completely acoustic phenomenon, and one that occurred only through multiple feedback-producing agents interacting in this decentralized manner. Different combinations of tones emerged over time based on the frequencies each vessel excited in one another.

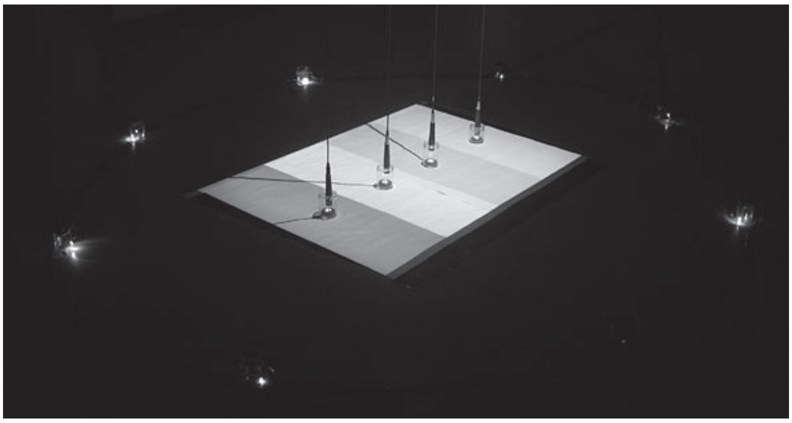


Figure 4. Image of *Filling Vessels* and *circadia* combined.

After developing these two interrelated sound installations, the idea of combining them became enticing. *Filling Vessels* and *circadia*, merged, forming a complex feedback-based environment. The small pulsations of *circadia* excited frequencies in the jars for *Filling Vessels*, coaxing unique balances between the small and busy activity of *circadia* and the enormous waves of sound emerging from the larger speakers in *Filling Vessels*. Images of light reflected in the installation space photographed by Tom O'Doherty were used for illumination, and processedz based on the activity of the four vessels in the space. The contrasting amounts of both light and sound affected the sense of the depth of the sound in space, making it seem both vast and localized.

4. Conclusion

Filling Vessels and *circadia*'s development utilize different implementations of self-regulating feedback systems. Both monitor the amount of amplification of the microphones inserted in the vessels as a way of controlling the amount of feedback that emerges. However, each utilizes adaptive games differently. One defines the responsibilities of the feedback-producing agents such that they have the goal of generating feedback, but only so long as it refrains from overpowering the other agents, while the other endows the agents with the task of synchronizing with one another. The self-regulating mechanism employed in both installations enabled their contrasting sonic characters. One criticism of both installations relates to the nature of the oblique interaction with the audience. Many visitors could sense some sort of influence on the system, but the immediacy of a cause and effect relationship was often difficult to pinpoint. This, however, is one of the aspects of interaction that *Filling Vessels* and *circadia* is concerned with. There are no knobs to turn, no buttons to push, no switches to throw. It is simply by being there that one affects the space, becoming part of it and the resonating environment.

Acknowledgements

Both *Filling Vessels* and *circadia* been developed with the generous support of the Studio für Elektroakustische Musik und Klangkunst of the Universität der Künste, Berlin, Culture Ireland, the Fulbright Kommission in Berlin through the NYSCA-sponsored artist residency program at Rensselaer Poltyechnic Institue's iEAR Studios, and the Music Department at New York University-GSAS. Enormous gratitude to Tom O'Doherty, who helped develop many visual elements of the installation. Many thanks to Martin Supper, Peter Castine, Elizabeth Hoffman, Pauline Oliveros, amoebazoid, William Lindsay, and vodstrup for their consistent and invaluable feedback. All photos for this paper, excluding the diagram from Figure 1, were taken by Tom O'Doherty.

References

Lucier, A. (1997) "Empty Vessels", http://www.ps1.org/cut/volume/lucier.html, July.

Lucier, A. Reflections: Interviews, Scores, Writings, MusikTexte, 1995.

Miranda, E. 2003. "On the Music of Emergent Behavior: What Can Evolutionary Computation Bring to the Musician?" *Leonardo*, 36(1): 55-59.

Um sistema de recomendação de músicas brasileiras

Maiara B. Monteiro[1], Natan R. Machado[1], Paulo A. L. Nogueira[1], Thales A. Campos[1], Fernando W. Cruz[1], Edilson Ferneda[1]

[1] Universidade Católica de Brasília
SGAN 916, Módulo B 71.790-160 Brasília DF

maiara_bsb@hotmail.com, natanucb@gmail.com, pbao.nog@gmail.com, thales_acampos@hotmail.com, fwcruz@ucb.br, eferneda@pos.ucb.br

Resumo. *Este artigo propõe uma base de metadados de músicas brasileiras e um sistema de recomendação musical associado. A base de metadados é fruto de uma revisão do modelo de dados do MusicBrainz e permite a identificação de* tracks *por meio de um índice de* fingerprint*. A recomendação musical proposta é precedida pela extração de características das músicas e auxiliada por um classificador de músicas por similaridade de gênero. Na versão corrente, o sistema consegue promover recomendações com um nível de acerto na ordem de 70 a 80 por cento.*

Abstract. *This article proposes a database of metadata of Brazilian music and a associated musical recommendation system. The basis of metadata is the result of a review of the MusicBrainz's data model and allows the identification of tracks by an index fingerprint. The musical recommendation proposed is preceded by the extraction of music characteristics and assisted by a binder of music by similarity of gender. In the current version, the system can promote recommendations with a level of accuracy in the order of 70 to 80 percent.*

1. Introdução

Constata-se, atualmente, uma mútua influência entre o aumento de buscas por músicas na Internet e a melhoria significativa das ferramentas de seleção, armazenamento, indexação e recuperação musical e da infra-estrutura de conexão. O vertiginoso crescimento nas vendas de música pela Internet vem gerando grandes mudanças na forma de disseminação e comercialização de músicas (Cruz, 2008). Isso tem feito com que cada vez mais artistas optem por ofertar diretamente na Web suas produções musicais, competindo em igualdade de condições com as grandes produtoras. Esse novo modelo de negócios favorece investimentos em estratégias de recomendação musical (Byrd, 2006) baseadas no comportamento do usuário, procurando prever seu gosto e suas vontades (Celma & Lamere, 2007).

Entre as diversas iniciativas para recomendação musical estão o Musicovery (www.musicovery.com), o Pandora (www.pandora.com) o One Lhama (www.onellama.com) e o last.fm (www.last.fm), que o fazem por meio de técnicas tais como *(i)* anotação das características do áudio e comparação de similaridades, *(ii)* levantamento de perfil comportamental do usuário e *(iii)* suas relações sociais. Em alguns desses sistemas, a eficácia dessas técnicas é testada disponibilizando-se gratuitamente músicas para o público, a fim de que opinem se um determinado conjunto de músicas que estão sendo tocadas possui algum nível de similaridade. Na prática, esses portais submetem o público a uma avaliação de resultados de classificação automáticos, ao mesmo tempo em que fornecem opções de compra das músicas ouvidas.

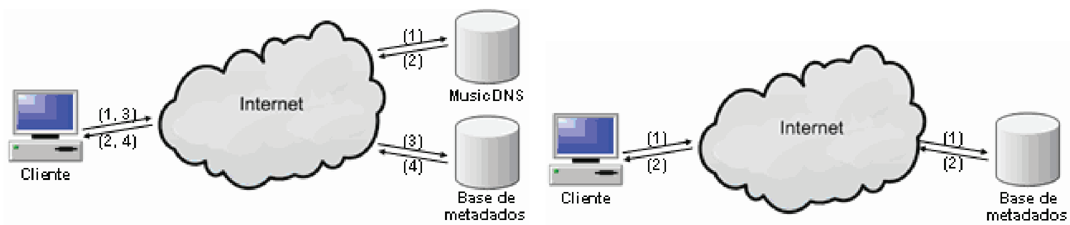
As alternativas citadas, apesar de também incluírem músicas brasileiras, têm um forte viés comercial e nem sempre atendem satisfatoriamente as demandas por produções musicais de caráter regional. Este trabalho descreve uma proposta de arquitetura de base de metadados de músicas brasileiras e uma estratégia de recomendação associada.

2. A base de metadados Estação Verde-Amarela (EVA)

A base de metadados sobre músicas brasileiras EVA é fruto de uma revisão do modelo de dados do MusicBrainz (musicbrainz.org), acervo de metadados disponibilizado na Internet, cujas tabelas contemplam informações sobre autores, títulos, entre outras. A base de metadados proposta é *opensource* e aproveita do MusicBrainz: *(i)* a modelagem de dados com as devidas alterações, *(ii)* o esquema de interfaceamento com os clientes, *(iii)* o mecanismo de moderação para controlar a validação dos metadados inseridos e *(iv)* o modelo ontológico subjacente. Ela diferencia-se do MusicBrainz por uma nova interface Web de manutenção dos metadados e pela remoção de tabelas desnecessárias à abordagem que está sendo adotada, em particular as que fazem uso do TRM, uma solução de *fingerprint* utilizada nas versões iniciais do MusicBrainz. No entanto, a maior modificação conceitual no projeto é relativa ao esquema de indexação e uso de *fingerprint*: apesar de ser uma base de metadados *opensource*, as informações do MusicBrainz são indexadas pelo PUID, um índice proprietário controlado pela MusicIP (www.musicip.com), o que limita o seu uso.

No MusicBrainz (Figura 1a), a base de metadados é acessada da seguinte maneira: **(1)** o cliente gera o *accoustic fingerprint* e o utiliza como parâmetro de busca no MusicDNS (www.musicip.com/dns/index.jsp), que **(2)** retorna o PUID associado ao *fingerprint*; **(3)** o cliente acessa o MusicBrainz, informando o PUID como parâmetro de busca; **(4)** o MusicBrainz retorna os metadados bibliográficos associados ao *fingerprint* em questão. A geração de *fingerprint* baseia-se no algoritmo OFA (MusicIP, 2006).

Este projeto investiga algumas alternativas para a geração do *fingerprint* segundo as recomendações de Cano (2004). Dentre as opções consideradas até agora estão o próprio OFA e o libFooID (www.foosic.org), aplicadas a um único formato musical (MP3). Além disso, prioriza-se a eliminação de uma solução proprietária para a identificação de *tracks*. Assim, o acesso à base de metadados se dá da seguinte maneira (Figura 1b): **(1)** o cliente MusicBrainz gera o *fingerprint* do arquivo musical, aplica um algoritmo de *hash* a este *fingerprint* e utiliza o resultado na consulta à base de *metadados*; **(2)** a base retorna os *metadados* bibliográficos associados ao *fingerprint* em questão.



(a) Arquitetura MusicBrainz *(b)* Proposta de acesso à base de metadados

Figura 1. Formas de acesso à base de metadados musicais

3. Esquema de recomendação adotado

A arquitetura proposta é baseada no modelo cliente-servidor. Estão sendo investigadas duas estratégias de recomendação. Na primeira, o cliente dispõe de uma interface de captura de amostras de áudio que são enviadas ao servidor. No lado servidor, é feita a análise do áudio coletado e a consequente geração de um rol de informações sobre músicas similares ao áudio recebido. Nesse caso, as recomendações ficam disponíveis aos usuários, assumindo que esses estejam previamente cadastrados no sistema de recomendação (como ocorre no last.fm). A Figura 2 representa essa estratégia.

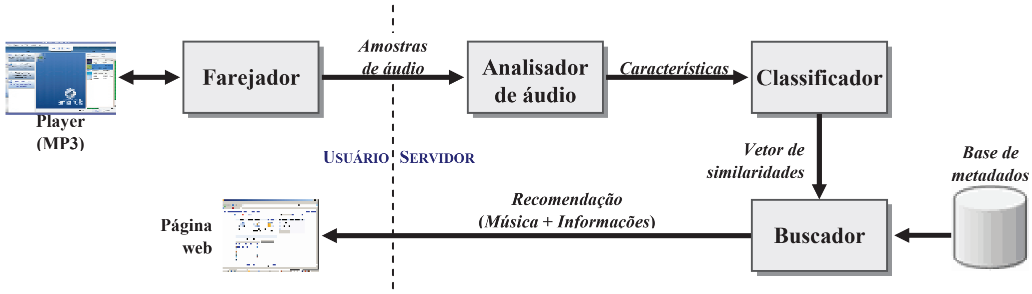


Figura 2. Estratégia de recomendação proposta

A recuperação das músicas é feita por um software "farejador", instalado na máquina do usuário, para recuperar amostras de áudio e enviá-las ao servidor. O servidor recebe as amostras, promove a recomendação e armazena as amostras colhidas para ampliação da base de músicas. A partir do áudio recebido, são extraídas suas características pela aplicação do Marsyas (http://marsyas.sness.net/), uma biblioteca voltada para análise de áudio. Essas características são submetidas a um classificador neural que produz um vetor de similaridades da música em relação aos estilos para os quais o classificador foi previamente treinado. Um módulo de pesquisa seleciona uma lista de músicas da base com as menores distâncias euclidianas em relação à música em questão. Essa lista, composta de informações recuperadas da base de metadados, é então disponibilizada para o usuário, que deverá estar previamente cadastrado para ter o direito de instalar o farejador em sua máquina e de visualizar as recomendações. Neste processo, está prevista a inclusão deste áudio na base, o que envolve o cálculo de seu índice de *fingerprint* e a geração de uma URI (*Uniform Resource Identifier*) associada.

O processo de construção do classificador foi fruto de uma série de experimentos. Nos testes iniciais, as amostras eram recebidas e o vetor de características internas de cada elemento era extraído com base nas experiências relatadas por Silla, Kaestner & Koerich (2007), considerando uma base composta inicialmente de 850 músicas. Como os resultados não foram satisfatórios, adotou-se uma nova estratégia com a composição de uma base de dados acessória com 200 dos melhores representantes de alguns dos principais estilos brasileiros. Essa base serviu como referência para a construção de um classificador capaz de identificar o grau de pertinência de uma determinada música em relação ao conjunto de estilos pré-determinados. Esse classificador é uma rede neural artificial do tipo *Multilayer Perception* (MLP), construída com o apoio do sistema Weka (http://www.cs.waikato.ac.nz/ml/weka/). A recomendação é feita comparando-se o grau de similaridade entre as características extraídas da música com cada uma das músicas que compõem a base de recomendações. O processo de seleção das características extraídas (*ZeroCrossings*, *Rolloff*, *Centroid*, MFCC, ...) foi feito empiricamente, ainda sem considerar qualquer fundamentação em técnicas de processamento de sinais.

Para melhorar o processo de recomendação, a estratégia inicial é combinada com uma segunda, baseada nas relações sociais do usuário e que usa o CoFE (http://eecs.oregonstate.edu/iis/CoFE/), um *framework* genérico de recomendação, para a filtragem colaborativa. Aqui, *(i)* o usuário atribui uma avaliação ao artista, expondo o seu grau de interesse; *(ii)* o sistema, com base no histórico dessas avaliações, busca por usuários com perfis semelhantes ao usuário alvo por meio do coeficiente de Pearson (Cruz, 2008), que calcula a correlação entre usuários; *(iii)* os usuários vizinhos são selecionados e, com base em seus perfis, é estimado o interesse do usuário alvo pelo artista.

4. Conclusões

Este trabalho apresenta uma base de metadados como suporte para um sistema de recomendação de músicas brasileiras. Na fase atual, a base de metadados é indexada por meio de uma solução de *fingerprint* que considera apenas arquivos no formato MP3, mas já está sendo investigada uma alternativa para que o sistema reconheça *tracks* em outros formatos.

As informações musicais consideradas neste trabalho são compostas por metadados bibliográficos armazenados em tabelas herdadas do MusicBrainz e pelos índices de características calculados com o auxílio de um classificador neural. Atualmente, por estar numa fase experimental, a base contém apenas 1.400 músicas, mas testes estão sendo feitos para um volume maior de informações.

Considerando a similaridade calculada a partir de informação extraída do áudio, foi possível fazer recomendações com acertos na ordem de 70 a 80% para quatro estilos musicais. A base está sendo revisada para a inclusão de novos estilos e serão consideradas, futuramente, outras estratégias de recomendação baseadas em características que envolvam o perfil do usuário e suas relações sociais. Em relação ao módulo cliente, na versão atual, o farejador interage apenas com o *player* Amarok (amarok.kde.org).

5. Referências

Byrd, D. (2006) “Other Topics in Music Organization and Representation (Lecture Notes)”. School of Music. Indiana University. <http://www.informatics.indiana.edu/donbyrd/INFO590Spring06/I590.html>.

Cano, P. (2006) “Content-Based Audio Search: from Fingerprinting to Semantic Audio Retrieval”. PhD thesis, Department of Pompeu Fabra University, Barcelona, Espanha. <http://mtg.upf.edu/files/publications/34ac8d-PhD-Cano-Pedro-2007.pdf>.

Celma, O.; Lamere, P. (2007) “Music Recommendation Tutorial”. ISMIR 2007, Viena, Áustria. <http://mtg.upf.edu/~ocelma/MusicRecommendationTutorial-ISMIR2007>.

Cruz, F. W. (2008) “Necessidades de Informação Musical de Usuários Leigos em Música”, Tese de Doutorado em Ciência da Informação, Universidade de Brasília.

MusicIP (2006) “Open Fingerprint™ Arqhitecture Whitepaper Version 1.0”. <http://www.musicdns.org/files/Open_Fingerprint_Architecture_Whitepaper_v1.pdf>.

Silla Jr, C. N., Kaestner, C. and Koerich, A. (2007) “Automatic music genre classification using ensemble of classifiers”, Proceedings of the IEEE SMC, Montreal, Canadá.

Considerations in the use of Computer Technology in Contemporary Improvisation

Are Computers Musical Instruments?

Cesar Villavicencio

Departamento de Música – Escola de Comunicações e Artes – Universidade de São Paulo

Email: cesar@cevill.com

Music has always had close ties with the development of instrumental technology. Although today computers hold an important role in performance practices, there are some issues regarding the physicality of musical presentations that need to be discussed. This paper focuses on some aspects involved in the use of computer technology today in the performance of music improvisation. Borrowing information from cognitive and aesthetic sources, it presents the idea of the necessity of developing instrumental interfaces for performing music with computer technology, arguing that sponsoring collaboration between scientific and performative areas would ultimately help to generate music that is eloquent and consistent.

Introduction

Soon after the introduction of tape as a musical medium, in the decade of the 1950's, the first experiments and compositions combining both tape and acoustic instruments began to appear. This was the beginning of live sound processing. Today, the technological possibilities are, needless to say, enormous. Computer technology is constantly becoming faster, more powerful and reliable. Yet, these qualities do not ensure the quality of musical performances that use electronic devices. Quite the opposite, they offer such broad possibilities that the danger of losing consistency is present as it probably never was before. I am of the belief that the physicality of musical performance is a crucial issue and a central aspect of the way music communicates in live situations.

When Hazel Smith and Roger Dean mention that "Computers can be both tools of improvisation and provide the environment (with)in which improvising takes place"[i], we also have to think that, since the role of a computer machine is not solely dedicated to the making of music, there are some issues that need to be addressed. Today, it seems that the lack of communication between the "laptop" performer and the audience is sometimes acknowledged as a new "concept" natural of this kind of performances. Although there might be unquestionable levels of expertise involved in using computers for performing music, I think that it is misleading to regard as inherent of this practice the lack of direct contact with the audience.

Body and Gesture

If we reflect on the idea that considers the body as playing a central role in musical activities, we should also think that there is an intrinsic corporeal engagement in musical communication, which allows moving sonic forms to affect our bodies. Although it maybe difficult to contextualize these musical corporeal significations, they exist and affect us in the process of listening to music (Leman, 2008). Roland Barthes in his *Musica Practica* acknowledges the existence of a "muscular music in which the part taken by the sense of hearing is one only of ratification, as though the body were hearing."[ii]. Furthermore, Richard Leppert tells us: "When people hear a musical performance, they see it as an embodied activity. While they hear, they also witness: how the performers look and gesture, how they are costumed, how they interact with their instruments and with one another."[iii] Conclusively, the observation of a performance has a dialectical perception that involves aspects related to the mind and body. We could also embrace the idea of gesture as a movement that can express something that embodies a special meaning (Iazzetta, 2000). In this matter, the focus concerns the physicality of live musical performances that use electronics in which regularly there is a divergence between what we listen and what we see. Richard Barrett, in a conversation, said that this lack of connection deprives audiences of understanding what is going on between the interpreter and the computer and creates an illusion that once it takes place, we run the risk that the perception of the music by the audience loses its connection with reality. I believe that we should not compromise in trying to adjust to this situation by understanding it as a problem that the audience needs to solve by adopting a different concept of listening, but rather to find manners to restore the vital connection natural of live performances.

Resistances

John Dewey's aesthetic theory reveals the importance of what he calls resistances to transform an impulse[iv] into an expression. An outward impulse without resistance would simply be a discharge. Dewey tells us that, "To discharge is to get rid off, to dismiss; to express is to stay by, to carry forward in development."[v] In theory, a resistance may manifest itself by meeting and overcoming the technical difficulties of, for instance, an instrument. Naturally, the learning process required for managing the electronic music's hardware and software could be thought as offering resistances as well. However, the manner for controlling them is usually restricted to a *mouse* or a couple of sensors, which do not offer a platform for developing the combination between the corporeal and the sound manipulation. Moreover, in the absence of predefined musical material, as in the case of contemporary improvisation, the resistances presented to the performer inherent in the process of interpreting a score, for instance, are absent. Hypothetically, the necessary resistances for composing the music in this environment depend on the intersubjective process of the collective artistic activity is which the performer is immersed. Now, if we apply to music making with computer technology those three concepts, (a) the intrinsic intercorporeal level of music performances, (b) the congruence between gesture and sound, and (c) the transformation of impulse into expression through the interpolation of resistances, it might be possible to move towards the direction of making music with computers an experience in reality for both performers and audiences.

The Midified Contrabass Recorder

One possible strategy to improve the physical experience of electronic music is to think of whether we can make the computer into a musical instrument. During the process of developing my interactive contrabass recorder, in collaboration with sonologists, technicians and composers at the Institute of Sonology in The Hague, issues such as the connection between the electronic sounds and the gestures were taken into consideration. We had to choose, for instance, where the sensors ought to be built according to the function they were intended to perform. Also, regarding the contact with the audience and the interactivity with co-participants, we came to regard the graphical interface as one of the main obstacles in the communication process. Relying on having the computer screen in front of the instrument induced me to look unnecessarily at the screen for most of the time during the performance. At STEIM (Studio for Electro Instrumental Music), in Amsterdam, we incorporated a small LCD monitor into the recorder so as to get the necessary information when needed while playing. By no longer having that distractive element in front of me and the instrument, a better connection with the audience emerged from which I felt a significant change in the music I played, one that was more eloquent and in tune with the co-participants and the public. What actually changed was the amount of time available for performing. Instead of using time for paying mostly unnecessary attention to the graphical interface, time was now being used for concentrating better on the reactivity of creating music.

Conclusion

Reflecting on the different roles music fulfils today, it is important to consider Mark Slobin when he suggests: "we need to think of music as coming from many places and moving among many levels of today's societies, just as we have learned to think of groups and nations as volatile, mutable substances rather than fixed units for instant analysis."[vi] Indeed, among the myriad of things that have changed due to technology we could affirm that it has made distances shorter, it has brought different concepts of time and helped us get in contact with different societies with which we can interact. I believe that the revolution brought by the use of technology in music still will prove to be one of profound changes in the manner we make and receive music, compared possibly to the changes once brought by Humanism. In my experience as a performer, I think that those profound changes in music could be brought to light more consistently by collaboration between experts in the areas involved in the performance with computer technology. It seems reasonable to think about abandoning traditional ideas such as the individual status of the "creator", the adulation of the performer, and the transcendence of musical creations in order to give space to a more pluralistic and collaborative endeavours in which all the parties bring the best of their field into the creation of music manifestations that moves our mind and body in deep enjoyment. Bringing performers, sonologists and engineers in partnerships with the objective of developing tools would maybe allow us to create unique instruments that are sophisticated and flexible, that are able to generate superior sound quality, and that let us be expressive and powerful. Once we start developing new instruments we also start a process of learning and discovering the richness of true artistic experiences of intellectual depth and expression immersed in our present cultural reality.

Notes

[i] Smith and Dean (1997) p. 249
[ii] Barthes (1977) p. 149
[iii] Leppert, R. (1995) p. xxii
[iv] In Dewey's own terminology impulses are referred to as *impulsions*, which "are the beginnings of complete experience because they proceed from need." Dewey (1934) p. 58.
[v] Dewey (1934) p. 62
[vi] Slobin, Mark (1993) p. xv

References

Barthes, Roland (1977) "Musica Practica" in: Image – Music – Text, Hill and Wang, New York, USA. pp. 149-154.

Chadabe, Joel (1997) "Electric Sound. The Past and Promise of Electronic Music" Prentice Hall, New Jersey, USA.

Daniel Fischlin and Ajay Heble (2004) "The Other Side of Nowhere. Jazz, Improvisation and Communities in Dialogue", Wesleyan University Press, USA.

Dewey, John (1934) Art as Experience, Penguin, New York, USA.

Iazzetta, Fernando (2000). "Meaning in Musical Gesture", in: Trends in Gestural Control in Music. Edited by M. M. Wanderley and M. Battier, Ircam – Centre Pompidou, France.

McComiskey, Bruce (2002) "Gorgias and the New Sophistic Rhetoric" Southern Illinois University Press, USA.

Leman, Marc (2008) "Embodied Music Cognition and Mediation Technology", MIT Press, Cambridge, USA.

Leppert, Richard (1995) "The Sight of Sound: Music, Representation, and the History of the Body. University of California Press, Berkeley, USA.

Slobin, Mark (1993) "Subcultural Sounds, Micromusics of the West" Wesleyan University Press, New England, USA.

Smith, Hazel and Dean, Roger (1997) "Improvisation, Hypermedia and the Arts Since 1945" Harwood Academic Publishers, Amsterdam, The Netherlands.

Villavicencio, Cesar (2008). "The Discourse of Free Improvisation: A Rethorical Perspective on Free Improvised Music" PhD Thesis. http://cevill.com/phdcesar

White, Eric Charles (1987) "Kaironomia: On the Will-to-Invent" Cornell University Press, New York, USA

Mapeamento Sinestésico: do Gesto ao Objeto Sonoro

José Fornari[1], Mariana Shellard[2], Jônatas Manzolli[1,2]

[1]Núcleo Interdisciplinar de Comunicação Sonora - NICS
[2]Instituto de Artes - IA

University of Campinas - UNICAMP

tutifornari@gmail.com, marianashellard@gmail.com,
jonatas@nics.unicamp.br,

Abstract. *This article presents a synesthetic mapping of conceptual drawings into sonic objects. The digital image of each drawing is transformed in sound using a image process developed in research. Resulting sonic objects were used to compose an interactive and multimodal sound installation.*

Resumo. *Este trabalho trata de um mapeamento sinestésico de desenhos conceituais em objetos sonoros. A imagem digital de cada desenho é transformada em som através de um algoritmo de processamento de imagens que foi desenvolvido na pesquisa. Os objetos sonoros foram utilizados na composição de uma instalação sonora multimodal.*

1. Introdução

O trabalho aqui reportada é inspirada na atividade artística dos músicos da chamada "Escola de Nova York", onde artistas como Morton Feldman, John Cage, Earle Brown e Steve Reich, criaram um pensamento que ressaltou a noção de processo na concepção musical. Do ponto de vista da nossa pesquisa, esse grupo de artistas criou as bases da Arte Generativa e nos parece importante pesquisar o uso do computador como ferramenta para criar processos generativos. A nossa segunda motivação é vinculada ao conceito de Sinestesia: condição sensorial peculiar quando um indivíduo, ao receber um estímulo em uma modalidade sensorial, imediatamente o percebe como um estímulo advindo de outro sentido (como ouvir uma cor ou ver um som). O estudo da sinestesia vem sendo desenvolvida no Instituto Prometeus, criado em homenagem a Scriabin (http://www.prometheus.kai.ru) trabalhos sobre esse tema são apresentados em [Campen,1999] e [Ahsen,1997] .

Este artigo descreve o desenvolvimento de uma obra artística sinestésica que se baseia no mapeamento de desenhos conceituais em objetos sonoros. A obra consiste em uma instalação artística onde as paisagens sonoras são geradas por um sistema computacional adaptativo descrito em [Fornari,2008] e relacionado com o modelo desenvolvido em [Moroni,2006].

2. Dos Desenhos aos Objetos Gráficos e Objetos Sonoros

Similar à partitura musical, um desenho pode também registrar uma informação artística: o gesto. Um desenho projeta-se assim em outras linguagens, como um processo de registro artístico, que possui significado por si só. Conforme descrito por Richard Serra, este processo é "uma forma de ver dentro de sua própria natureza. (...) Não existe uma maneira de se fazer um desenho, existe apenas o desenho" [Serra,1994].

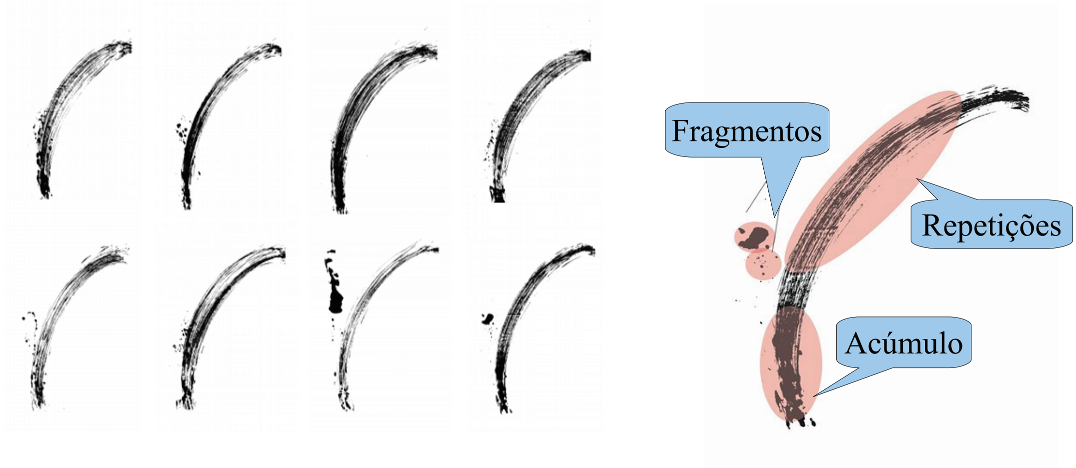


Figura 1. Imagem digital de 8 desenhos gestuais da série de 380 desenhos (esquerda) e as indicações das 3 características gráficas estabelecidas (direita).

Os desenhos analisados durante a pesquisa foram concebidos como sendo o registro de um gesto, no qual pretendia-se analisar as variações e transformações de um movimento delimitado, ao longo de um determinado período de criação da obra (aproximadamente dez meses). Ao todo, foram criados cerca de 380 desenhos e cada um deles foi desenhado num período de dez a trinta segundos usando nanquim sobre papel-filtro. Devido à característica aquosa da tinta associada à alta absorção desse tipo de papel, criaram-se também respingos e acúmulos de tinta conforme a intensidade e orientação do movimento. A Figura 1 (esquerda) mostra as imagens de oito desenhos da série de 380 desenhos.

O mapeamento partiu da identificação de aspectos gráficos que, posteriormente, foram associados com aspectos sônicos. Essa análise partiu de três características do desenho e do material (tinta nanquim e papel-filtro) são elas: **1) acúmulo, 2) repetições, 3) fragmentos.** Para cada uma das três característica, desenvolvemos seus equivalentes sônicos que representam sinestesicamente a projeção dos desenhos no domínio sonoro.

A Figura 1 (direita) mostra um exemplo de imagem de um desenho (esquerda) com as três características, acima estabelecidas, grafadas sobre essa imagem (direita). Em nossa classificação, cada desenho possui apenas um **acúmulo**, que é a região de maior concentração de tinta, normalmente associada ao início do gesto na região inferior esquerda. As **repetições** são traços localizados na região central, onde o gesto era mais determinado e reto. Os **fragmentos** são dados pelas áreas de respingo de tinta, destacados do acúmulo. São manchas aproximadamente circulares e aleatoriamente criadas.

Foi desenvolvido um algoritmo para o reconhecimento dos objetos contidos em cada imagem digitalizada. Esse mapeia cada desenho em distintos objetos gráficos que consistem em um único acúmulo, diversas repetições e diversos fragmentos. Na Figura 2 tem-se a seqüência de imagens do processamento feito por este modelo. O segundo valor de cada objeto refere-se à uma métrica que descreve o grau de circularidade deste objeto.

3. Resultados

O algoritmo encontrou 35 objetos no desenho 13 apresentado na Figura 1 (direita). A Figura 2 mostra um detalhe ampliado dessa análise, onde pode-se observar com mais precisão a região de contorno dos objetos 1, 2 e 8, dado por um contorno em branco, e os seis parâmetros coletados do objeto 2.

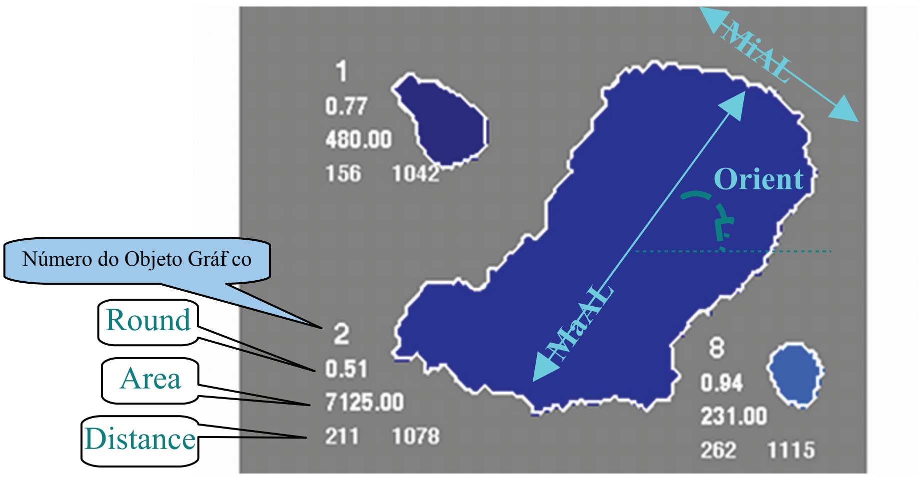


Figure 2. Detalhe dos objetos gráficos: 1, 2 e 8 da imagem do desenho 13 da Figura 1 (direita).

A Tabela 1 mostra a correspondência entre as características escolhidas para analisar cada desenho e a associação entre elas e os aspectos sonoros relacionados no algoritmo de mapeamento.

Tabela 1. Os objetos gráficos mapeados dos desenhos e a correspondência entre suas características gráficas e aspectos sonoros

Objeto Gráfico	Característica gráfica	Aspecto Sonoro Relacionado
Acúmulo	Concentração maior de tinta na base do desenho, onde se iniciou o gesto	Ruído de baixa freqüência e constante
Repetição	Traços gerados pelo movimento repetitivo do gesto	Tons, variação de altura e freqüência
Fragmento	Respingos de tinta, decorrentes da intensidade do movimento	Pulsos de curta duração, variando do ruidoso ao tonal

A Tabela 2 mostra os seis parâmetros coletados pelo algoritmo, sobre cada objetos encontrado pelo mapeamento de uma imagem de um desenho da coleção.

Tabela 2. Os seis parâmetros coletados de cada objeto gráfico.	
Area	área de cada objeto encontrado pelo mapeamento. A medida é feita pela quantidade de pixels de cada objeto.
Round	circularidade de cada objeto, dado pela métrica da Equação 1. Os objetos mais circulares (round < 0,5) são fragmentos, correspondendo aos sons de curta duração, enquanto que os menos circulares (round > 0,5) são repetições, correspondendo aos sons contínuos e tonais.
Orient	ângulo de inclinação de cada objeto, a partir do eixo horizontal da imagem mapeada.
Distance	distância entre o centro de gravidade de cada objeto e a origem da imagem mapeada. Este parâmetro é utilizado para ordenar temporalmente o início dos sons correspondentes aos fragmentos e repetições.
MaAl	Medida da maior extensão do formato de cada objeto.

MiAl	Medida da extensão mínima do formato de cada objeto. Se o objeto é circular, MaAL e MiAL são idênticos.

Cada objeto gráfico, encontrado pela análise computacional, corresponde a um objeto sonoro de tal forma que os aspectos gráficos estejam presentes e perceptualmente evidenciados no som resultante. O acúmulo foi associado à escala de longa duração. A repetição refere-se à escala de média duração. E os fragmentos referem-se à escala de curta duração. Seguindo esta classificação, definimos parâmetros básicos para serem coletados automaticamente pelo modelo computacional de cada objeto gráfico mapeado dos desenhos. Foram escolhidos os seis parâmetros que estão descritos na Tabela 2.

Os valores normalizados dos seis parâmetros, calculados automaticamente pelo algoritmo, para os três objetos mostrados na Figura 2, são apresentados na Tabela 3.

Tabela 3. Valores normailzados dos aspectos dos objetos.

Aspectos \ Objeto	Area	Round	Orient	Distance	MaAL	MiAL
1	0.0026	0.64	-0.61	0.53	0.0164	0.0419
2	0.0379	0.42	0.56	0.55	0.0693	0.1515
8	0.0012	0.78	-0.97	0.58	0.0096	0.0330

Note que os valores de *Round* e *Area*, da Tabela 3 não coincidem com os valores mostrados na Figura 3. Isto ocorre porque os valores da Tabela 3 foram normalizados em relação aos valores encontrados para os 35 objetos. De posse desses dados, objetos sonoros foram criados e os resultados sonoros de tais mapeamentos encontram-se disponíveis no link: http://www.nics.unicamp.br/~fornari/sbcm09.

4. Referências

[Campen,1999] Chrétien van Campen. "Artistic and Psychological Experiments with Synesthesia," Leonardo, 32, 1 (1999), pp. 9-14, ref. on p. 10.

[Ahsen,1997] Ahsen, A. "Visual imagery and performance during multisensory experience, synaesthesia and phosphenes," Journal of Mental Imagery 21, 1-40 (1997)

[Fornari,2008] Fornari, J.; Mnzolli, J.; Maia Jr. A. *Soundscape Design through Evolutionary Engines*. Special Issue of "Music at the Leading of Computer Science". JBCS - Journal of the Brazilian Computer, 2008.

[Moroni,2006] Moroni, A. Manzolli, J. von Zuben. ArTbitrating JaVox: evolution applied to visual and sound composition. In: Proceedings of SIACG'2006-Ibero American Computer Graphics Symposium, Santiago de Compostela, Espanha. (2006).

[Serra,1994] Serra, Richard. *Writings Interviews*. The University of Chicago Press, Chicago, 1994.

Um *External* de Aspereza para Puredata e MAX/MSP

Alexandre Torres Porres[1], André Salim Pires[2]

[1]Departamento de Música / [2]Instituto de Matemática e Estatística – USP

Universidade de São Paulo (USP) Rua do Matão 1010 Cidade Universitária 05508-900 - São Paulo, SP - Brasil - Caixa-Postal: 66281

porres@gmail.com, apires@ime.usp.br

Abstract. *This paper reports the development of a Puredata & MAX/MSP External that measures the Roughness of a sound spectrum. Technical details of implementation, theoretical investigation e musical aplications are presented on this text, as well as further and parallel research.*

Resumo. *Este artigo reporta o desenvolvimento de um "external" para Puredata e MAX/MSP que mede a Aspereza de um espectro sonoro. Detalhes técnicos de implementação, investigação teórica e aplicações musicais são apresentados, assim como desenvolvimentos paralelos de futuros.*

1. Modelos de Aspereza

O termo psicoacústico *Roughness* é traduzido como Aspereza, mas, devido a uma certa falta de consenso e padrão, também é possível encontra-lo traduzido como Rugosidade. O termo é uma analogia à sensação tátil, que se refere à percepção de pequenas irregularidades no som. O correlato físico da Aspereza (tal como o de Batimentos) é a *Flutuação de Amplitude* [Vassilakis 2001]. As flutuações lentas, abaixo de 20Hz, são percebidas como tremolo, enquanto que as acima dessa taxa (até um intervalo que depende da Banda Crítica) promovem a sensação da Aspereza. As flutuações de amplitude (ou Modulação de Amplitude) podem ocorrer pela simples superposição de dois tons puros, onde a taxa de flutuação é igual à diferença entre os tons.

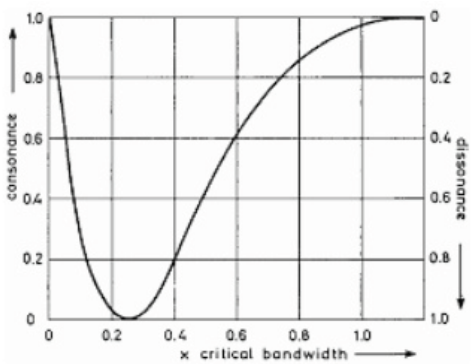


Figura 1. Aspereza de tons puros, segundo Plomp e Levelt (1965).

O estudo clássico de medição da Aspereza é de Plomp & Levelt (1965), que relacionaram essa sensação com a Banda Crítica, tal qual medida por Zwicker (1961). A máxima do estudo de Plomp & Levelt é que a sensação mais forte de Aspereza para dois tons puros ocorre quando eles estão em um intervalo que corresponde a um quarto da Banda Crítica. Na Figura 1, temos o gráfico de Aspereza em função da diferença de dois tons puros na escala *Bark*, que é a unidade da escala das Bandas Críticas.

Uma série de modelos foram desenvolvidos baseados nesses dados essenciais, e a grande diferença está em como eles contabilizam a Aspereza para tons puros com amplitudes diferentes. Vassilakis (2001) indica que o estudo de Terhardt (1974) tem sido negligenciado na literatura sobre Aspereza, e, a partir dele, desenvolve um modelo de como contabilizar a rugosidade a partir de dois valores de amplitudes distintos. Clarence Barlow (2008) não adota esse método, mas se preocupa em adotar medidas de intensidade em *Phons* e *Sones*, por meio de uma fórmula de conversão que ele desenvolveu a partir das curvas originais de Fletcher e Munson (1933).

2. O Modelo de Aspereza do External

O primeiro autor já publicou um estudo sobre um modelo original de Aspereza que combina elementos de vários modelos estudados [Porres, Manzolli 2007a], e já desenvolveu um *Patch* de Puredata que implementava esse modelo com algumas aplicações musicais [Porres, Manzolli 2007b]. A partir do mesmo estudo, os autores desenvolveram um código em C para gerar objetos de Pd e Max. O modelo original de Aspereza do primeiro autor incluía fórmulas de Parncutt (disponível em <http://www-gewi.uni-graz.at/staff/parncutt/rough1doc.html>) e Sethares (2005) para modelar o resultado de Plomp e Levelt, com a ressalva que a fórmula de Parncutt é mais acurada.

O modelo se assemelhava em muito ao de Clarence Barlow (2008), com a diferença da inclusão do estudo de Vassilakis (2001), e em deixar de lado a abordagem de Barlow para ter um peso que influencia na medida de Aspereza, que consiste em extrair a média geométrica dos *Sones* de um par de senóides. No recém criado *external*, tivemos, a princípio, os mesmos elementos que no *Patch*. Inserimos parâmetros de inicialização para que seja possível alternar entre a curva de Parncutt e Sethares, e preferimos também incluir o método de Barlow integralmente, para que seu modelo original pudesse ser comparado com a revisão deste estudo. De tal modo, o argumento *0* indica a fórmula de Parncutt, o argumento *1* a de Sethares, e o argumento 3 é o modelo de Barlow, que adota a curva de Parncutt e sua média geométrica de *Sones*.

Um passo seguinte foi a inclusão de um modelo de mascaramento, que redimensiona ou zera amplitudes do espectro. A fonte foi o modelo de *Pitch Commonality* de Parncutt, disponível em: <http://www.uni-graz.at/~parncutt/computerprograms.html>.

4. Aplicação do Modelo

Dentre tantos parâmetros, surgem perguntas importantes. Por exemplo: Quão essenciais são elementos como o mascaramento? Por essa ser uma inclusão nova no modelo, tal questionamento ainda não foi propriamente investigado. Tampouco chegou-se a uma conclusão de que tal abordagem de modelar o mascaramento seria a ideal. Antes da inserção do mascaramento, o modelo, ainda em formato de *Patch* de Puredata, já foi aplicado pelo primeiro autor em análise de Curvas de Afinação. Onde, dado um espectro, podemos gerar curvas de Aspereza para essa sonoridade em intervalos musicais distintos. Pontos mínimos são resultados de alinhamento de parciais, e podem ser relacionados com a sensação de afinação, ou de um intervalo consonante (Vide Figura 2). Em contrapartida, pontos máximos podem se relacionar a uma percepção de desafinação/dissonância. Em todo caso, antes de mais nada, há de haver um contraste, uma diferença razoável entre pontos máximos e mínimos da curva, para que tais sensações de consonância/dissonância sejam consideradas relevantes no trecho em questão. Por exemplo, na figura abaixo, um alinhamento de parciais (vale) na região do Trítono é bem menos relevante que na região da Oitava.

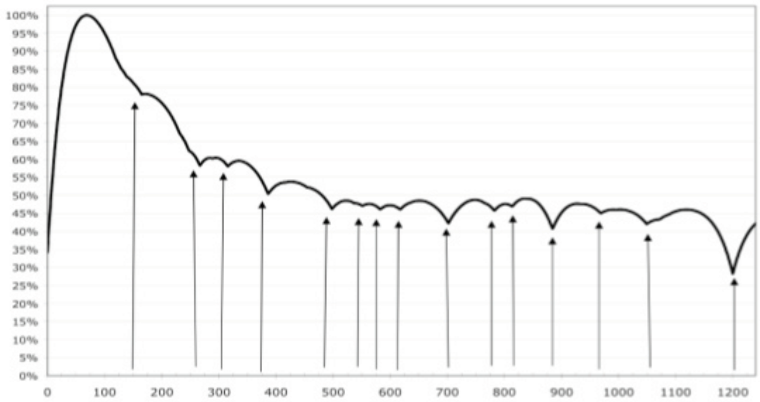


Figura 2. Curvas de Afinação. Aspereza no eixo vertical e cents no horizontal

O elemento de mascaramento está longe de ser descartado neste estudo, uma vez que a Aspereza depende da audibilidade de parciais. Esse ainda é um trabalho não muito bem resolvido. Segundo Parncutt em correspondência eletrônica particular, apesar de tantos modelos que têm sido desenvolvidos há décadas, ainda não há um modelo de Aspereza que tenha sido desenvolvido e aplicado musicalmente de modo eficiente e interessante.

Outros modelos de Aspereza possuem abordagens diferentes e são aplicados para fins não musicais, como a medição de ruído em automóveis. Cabe a essa pesquisa aplicações musicais, em especial as que relacionam conceitos subjetivos, como a dissonância, a atributos perceptivos psicoacústicos, como a Aspereza.

5. Desenvolvimentos Paralelos e Futuros

Faz parte desta pesquisa discutir a fundo diversos detalhes técnicos e confrontar dados de diversos modelos. Este estudo contribui ao revisar os modelos de Aspereza baseados em Plomp e Levelt (1965). Dentre outros modelos com abordagens diferentes, um caso especial ainda a ser estudado é o trabalho de Leman (2000). O trabalho de Pressnitzer e McAdams (1999) é outro que falta debater melhor sobre, por também possuir uma implementação em Pd. Entretanto, sua abordagem analítica é bem distinta, dificultando a relação com este estudo e implementação. Mas de acordo com Vassilakis (2001), ignorar essa questão da fase não distorce significativamente os resultados.

O External desenvolvido para Puredata e MAX/MSP envolve muitos passos, como rotinas de conversões acústicas e psicoacústicas, a exemplo da conversão entre Hertz e *Barks*, e de dB para *Phons*/*Sones*. O modelo de mascaramento deve ter um certo destaque, por ser útil para outros fins, a começar pelo modelo de *Pitch Commonality* de Parncutt, de onde originalmente foi extraído. A intenção é que não somente um *External* resulte dessa pesquisa, mas sim uma biblioteca, como é parte do escopo maior da pesquisa de doutorado do primeiro autor, que coincide em alguns pontos com a pesquisa do segundo autor.

Em vista de computar as Curvas de Afinação, o modelo de Aspereza pode ser concatenado com ainda outros Modelos e atributos psicoacústicos, como harmonicidade do espectro, relacionado a modelos já existentes, que devem fazer parte da biblioteca a ser desenvolvida em pesquisa. Uma das possíveis aplicações musicais pode ser a de analisar um som em tempo real, extrair parâmetros que correspondam com os de outros sons, para que então sejam disparados na seqüência. Tais sons podem ser previamente gravados, ou captados da própria situação de performance ao vivo.

Os *Externals* e publicações antigas do primeiro autor podem ser obtidos em seu site: <http://porres.googlepages.com>. A implementação em outros ambientes também seria do interesse desta pesquisa, se possível.

6. Agradecimentos

Barlow tem compartilhado arquivos pessoais, e sempre foi solícito em atender diversas dúvidas. Vassilakis, Sethares e Parncutt foram também muito prestativos. Esta pesquisa é financiada por bolsa do CNPq, e também recebe apoio da FAPESP por estar vinculada ao projeto temático MOBILE: Processos Musicais Interativos, na USP.

Referências

Barlow, K. (2008). Von der Musiquantenlehre (2 volumes, 130 páginas, em alemão) Feedback Paper 34, 2008. Feedback Studio Verlag, Colônia, Alemanha

Fletcher, H. e Munson, W.A. (1933), "Loudness, its definition, measurement and calculation" in *Journal of the Acoustical Society of America* N°5, 82-108 .

Leman, M. (2000), "Visualization and Calculation of the Roughness of Acoustical Musical Signals Using the Synchronization Index Model (SIM)," Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00),pp.125-130,Verona,Italy,December, 2000.

Plomp, R. e Levelt, W.J.M. (1965). Tonal Consonance and Critical Bandwidth. *Journal of the Acoustical Society of America* N° 38, 548-568.

Porres, T.A., Manzolli, J. (2007a) Um Modelo Psicoacústico de Rugosidade. In Proceedings of XI SBCM 2007, São Paulo.

Porres, T.A., Manzolli, J. (2007b) A Roughness Model for An Adaptive Tuning Patch Controlled by Antennas. In Proceedings of the 2nd International Puredata Convention 2007, Montreal.

Pressnitzer D. McAdams, S (1999). "Two phase effects in roughness perception," *J. Acoust. Soc. Am.* 105(5): 2773-2782. 1999.

Sethares, W.A. (2005) *Tuning, Timbre, Spectrum, Scale*. London: Springer-Verlag. 2ª Edição com CD-ROM.

Terhardt (1974) On the Perception of Periodic Sound Fluctuations (Roughness): Acustica, 30, 201-213.

Vassilakis, P.N. (2001) Perceptual and Physical Properties of Amplitude Fluctuation and their Musical Significance. *Dissertação de Doutorado*. UCLA.

Zwicker, E. (1961) Subdivision of the audible frequency range into critical bands (Frequenzgruppen), *Journal of the Acoustical Society of America*. *33*: 248.

Towards a Genetic L-System Counterpoint Tool

Bruno F. Lourenço , José C.L. Ralha , Márcio C.P. Brandão

[1] Departamento de Ciência da Computação
Universidade de Brasília (UnB) – Brasília, DF – Brazil

{brunofigueira,ralha,brandao}@cic.unb.br

Abstract. *In this article we discuss some aspects of algorithmic composition with L-Systems and how it can be enhanced with genetic operators. We attempt to create counterpoint with Genetic L-Systems and we present a few results and scores extracted from them.*

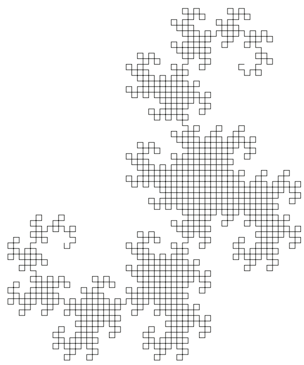
1. Introduction

L-Systems are rewritting systems first described by Aristid Lindenmayer [1]. They consist of an axiom and production rules that can be used to derive strings. Figure 1 shows an L-System for the famous dragon curve.

```
#level 11
#delta 90
#axiom FX
-------------------
Production rules

X -> X+YF+;
Y -> -FX-Y;
-------------------
Resulting strings
0 FX
1 FX+YF+
2 FX+YF++-FX-YF+
```

(a) L-system Grammar



(b) The Dragon Curve after 11 iterations

Figure 1: Dragon curve.

The work of Prusinkiewicz [1, 2] and others established L-Systems as tools for graphical modelling of objects that exhibit auto-similarity such as flowers, trees and fractals. The usual technique to render an L-System is to interpret each symbol in a LOGO-like manner. For such approaches, "F" means draw a segment with length d, "+" means turn the turtle +δ degrees, "-" means turn the turtle -δ degrees. "X" and "Y" are just auxiliary symbols and do not have a graphical interpretation.

In order to extract a score or a melody from the strings produced by an L-System we must use a certain musical rendering. Prunsiecwicz [3], for example, described a *spatial rendering* where a score is derived from the graphical interpretation of an L-System by projecting it on a musical scale. Each horizontal segment of the picture is interpreted as a note with a length proportional to the length of the segment. The pitch of a note is the y-th note of the chosen musical scale, where y is y-coordinate of the segment.

Other authors have sought to separate the graphical interpretation from the musical one. In [4], the authors described two techniques: the *sequential* and the *schenkerian* rendering. Both techniques do not need to go through the graphical interpretation to extract melodies. There is also the work of McCormack [5, 6], where he described L-Systems that use notes (*A,B,C..,G*) instead of LOGO style commands (*F,+,-*), and devised some mechanisms to express polyphony.

But why bother to use L-Systems at all? Mason and Saffle [7], for example, showed that both traditional western music and music generated by L-Systems share the same degree of self-similarity, so it's possible and plausible to produce interesting melodies that sound "familiar" to western ears. In the same article they suggested that we could even create a feeling of counterpoint by reading different rotations of an L-System at the same time.

Following the steps laid by Mason, we explore counterpunctual possibilites of L-Systems, but instead of relying on different rotations of the same L-Systems, we explore other possibilities using Genetic L-Systems [8] as a way of adding variability.

2. Genetic L-Systems

Genetic L-Systems are described in details in [8], and the reader should refer to that article for further explanation. Basically, a Genetic L-Systems is an augmented L-System with mutation and crossover capabilities. For example, a Genetic L-System for the Dragon curve is shown in Figure 2 . Each time the symbol Y is replaced, the crossover operator is triggered and modifies rules 0 and 1 and the same goes for X, but the mutation operator modifies rule 1 instead. An important aspect of this technique is that there are no fitness functions as in most Genetic Algorithm approaches, and the reader should also refer to [8] for a discussion of this matter.

```
#axiom FX

X -> X+YF+,
Y -> -FX-Y,crossover(0,1)
--------------------------------------
        Resulting string
0 FX
1 FX+YF+
2 FX+YF++-FX-YF+
```

Figure 2: Genetic dragon curve

3. Counterpoint and L-Systems

Mason and Saffle [7] explored what happened when two rotations of the same L-System are played together and noticed that it created a feeling of counterpoint. With Genetic L-Systems we can make a few more combinations. We made three experiments joining: two mirror versions of the same Genetic L-System; two different realizations of the same Genetic L-System and finally different genetic versions of the same L-System. In all experiments we used the spatial rendering [3].

On our first experiment we did something similar to what Mason previously did, but instead of using rotations of the same L-System we created "mirrored" versions by changing $+$ for $-$ and vice-versa. So instead of rules X→X+YF+ and Y→-FX-Y, we have rules X→X**-**YF**-** and Y→**+**FX**+**Y.

Since we are using a mirrored version of the same L-System, we produce something close to what is usually called *first-species* counterpoint, because of its note against note structure. While the example of such counterpoint in Figure 3 does not sound bad at all, in this particular example it feels dull since we produce the same rhythm for both voices.

Figure 3: First experiment: Genetic Dragon Curve and its "mirrored" version

If we use different realizations of the same Genetic L-System, because of their stochastic nature we might produce more interesting scores, but obviously the intervals between notes will be unpredictable. Figure 4 shows an example of our second experiment based on this technique of joining different realizations of the same Genetic L-System. We used the dragon L-System described on Figure 2. Since both voices are independent the feeling of counterpoint is more evident. In this case, we didn't find dissonant intervals between both voices.

Figure 4: Second experiment - Two different realizations of the Genetic L-System shown in Figure 2 played together

On our third experiment, we explored the possibility of using different genetic operators for the same L-System. Figure 5 shows two realizations of the Dragon curve using two different genetic operators. The first L-System is shown in Figure 2, the second is almost the same but the crossover is "linked" to the first rule instead of the second one. So, in the first L-System the crossover operation is done each time the Y is replaced, while in the second the crossover operation is done each time the X is replaced.

4. Conclusion

In this work we investigate the possibilities of using Genetic L-Systems to generate counterpoint. By mirroring the L-Systems, we were able to create melodies that were reminiscent of first-species counterpoint because of its note against note structure. More independence between voices can be achieved by means of generating two different realizations of the same Genetic L-System. The feeling of counterpoint is obtained by playing both melodies together. Another possibility we explored was the use of the same L-System but with different genetic operators.

Figure 5: Third Experiment - Dragon Curve

The quality of a musical output is always highly subjective, but we think that the results were interesting. We intend to incorporate more counterpoint knowledge into the rules themselves, but we need to explore further this matter since the modelling of interaction between voices is essential to a good counterpoint. But even *without* this modelling we could achieve interesting results with fairly simple production rules.

References

[1] P. Prusinkiewicz and A. Lindenmayer, *The algorithmic beauty of plants*. New York: Springer-Verlag, 1990. [Online]. Available: http://algorithmicbotany.org/papers/#abop

[2] P. Prusinkiewicz *et al.*, "*Algorithmic Botany*," University of Calgary: http://algorithmicbotany.org/.

[3] P. Prusinkiewicz, "Score generation with L-systems," *Proceedings of the 1986 International Computer Music Conference*, pp. 455–457, 1986.

[4] P. Worth and S. Stepney, "Growing music: musical interpretations of L-systems," *Springer*, vol. 3449, pp. 545–550, 2005. [Online]. Available: http://www-users.cs.york.ac.uk/~susan/bib/ss/nonstd/eurogp05.htm

[5] J. McCormack, "Grammar based music composition," 1996. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.8731

[6] ——, "The application of L-systems and developmental models to computer art, animation, and music synthesis," Ph.D. dissertation, School of Computer Science and Software Engineering, Monash University, Clayton, Australia, 2003.

[7] M. Saffle and S. Mason, "L-Systems, melodies and musical structure," *Leonardo Music Journal*, vol. 4, p. 8, 1994.

[8] B. F. Lourenço, J. C. L. Ralha, and M. C. P. Brandão, "L-Systems, Scores and Evolutionary Techniques," in *Proceedings of the 6th Sound and Music Computing Conference*, 2009, pp. 113–118.

Process and methodology leading to the acquisition and analysis of Event Related Potentials with basic sound stimuli

Bruno Giesteira[1], João Travassos[2], Diamantino Freitas[3]

[1] FBAUP/LSS, [3] FEUP/LSS
Universidade do Porto

[2] ESTSP/CEMAH
Instituto Politécnico do Porto

bgiesteira@fba.up.pt, jpctravassos@gmail.com, dfreitas@fe.up

Abstract. *This poster (short paper) has the main purpose to systematize the current working progress study which evaluate the clinical concept of Evoked Potentials using for that visual and auditory stimuli reasoned in the communication visual field and the audiological area, where the stimuli usually used for the clinical tests were recovered in a large spectrum of sounds (Different frequencies, intensities and time with a pure tone sound).*

Our poster communicates the process and methodology that is currently being used to our team to catch similarities of brain response (energy; latency; waveform; et al.) between visual and auditory basic stimuli.

1. Evoked Potentials: Concept and Goals

Evoked Potentials can be used to evaluate cortical and subcortical structures of the brain, such as the visual and auditory cortex and their pathways, which are responsible for the senses of vision and hearing respectively [Chiappa 1997; Niedermeyer *et al.* 2004]. One type of evoked potentials, Event Related Potentials (P300 tests), are also, more recently, being used to evaluate some of the high level characteristics of information processing in the central nervous system [Hruby; Marsalek 2003], such as the cognitive responses of the individual, like his capacity to identify and discriminate a particular battery of stimuli. That gives us an opportunity to acquire electrical cerebral signal corresponding to the brain reaction of some visual and auditory basic properties (colors; frequencies; et al.) as well as more complex and superior mental concepts (scale; depth; movement; et al.) and, through this, infer some stimuli patterns and trends always reasoned in clinical guidelines like through the measuring of peak amplitude and latency (time interval between the presentation of stimulus and the onset of a given peak) [Chiappa 1997; Niedermeyer *et al.* 2004; Blum *et al.* 2007; Walsh *et al.* 2005].

The evoked potentials can be extracted from the background electrical activity through averaging techniques [Chiappa 1997; Blum *et al.* 2007; Walsh *et al.* 2005]. Since the electric manifestation of the brain when exposed to a given stimulus occurs in the same time interval every time the stimulus is presented, and considering that the rest of the electrical activity is random and is not associated to the stimulus, it is possible to extract the desirable signal (Evoked Potential) through the acquirement of one signal per stimulus presented. Then all the signals collected are averaged to suppress the background noise, supposedly random, and to show the evoked potential that is constant [Chiappa 1997; Niedermeyer *et al.* 2004; Blum *et al.* 2007]. For that we developed an algorithm in "*Matlab"* capable to average the manifold cerebral signal responses to visual and auditory stimuli that were presented in tests.

2. Battery of Stimuli: Grounds and Pleas

Our main purpose is to catch possible relations between the basic visual grammar [Dondis, D.; Arnheim, Rudolf; *et al.*] and the perception of unassuming sounds. For that we developed a cluster of images, which are viewed on a computer display, capable to translate objectively the fundamentals of the visual grammar, namely: The three primary light colors; Dot; Line; Texture; Dimension; Scale; Movement. For the auditory stimuli we selected an audiological grammar (mainly used on clinical exams) since that is scientifically accepted, translating to sound, as far as possible, the same basic visual concepts with the few sound parameters used to clinical purpose namely: Pure Tone with 2, 3, 4, 5, 12 KHz frequencies; Tone Burst; Square Tone; Intensity with -6, -12, 0 db; and Time Intervals.

Figure 1. Example of a visual stimulus: Depth (first level)

The correspondent auditory stimulus of the previous example image is a Pure Tone at 2 KHz varying the intensity of sound in 6db (-6db to -12db). One second with -6db and the same period of time with -12db.
These simple grammars provide us the opportunity to analyze and discuss possible cognitive correlations and differences between visual stimuli and also compares the results with the standard/clinical auditory stimuli, focusing in the morphology of the respective waves and energy generated (potential differences between the active and reference electrodes) [Blum *et al.* 2007].

Our current results in Visual Evoked Potentials already prove that different stimuli produce a different brain response, in a way that one can distinguish some of those stimuli solely based on evoked potentials acquisition signal. We expecting with this similar methodology getting analogous results to the Event Related Potentials (P300 tests).

3. Process and Methodology: Signal Acquisition

All records are acquired using the "*Biopac Systems mp 150*" hardware with the EEG module - "*EEG100C*" - associated to it. The EEG module is set with a gain of 5000, a high pass filter of 0.1Hz, a low pass filter of 35Hz and a Notch filter (50Hz). The software used for the acquisition is "*AcqKnowledge 3.9.0*". All stimuli are processed and presented via "*SuperLab 4.0*" software. The "*AcqKnowledge*" and "*SuperLab*" software are installed on different computers. Averaging techniques are only applied after acquiring the signal with "*MatlabR2008B*".

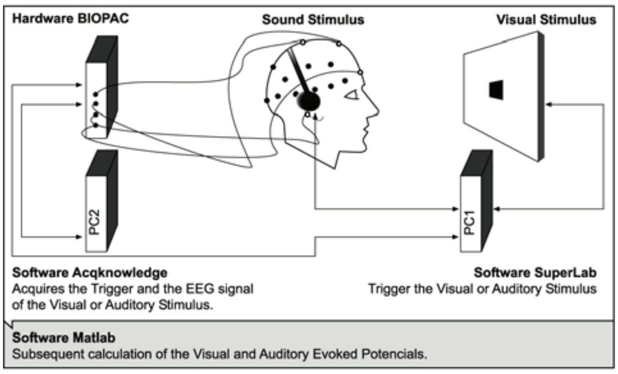


Figure 2. Signal Acquisition Schema

For this study we use two small reusable shielded electrodes - "*EL254s*" - for the two active electrodes, two general purpose shielded electrodes – "*EL258s*" – for the two references and one general purpose unshielded electrode - "*EL258RT*" - that serve as the ground electrode. Three channels are set, two analog channels for the EEG signal and one digital channel for the stimuli input/onset from "*SuperLab 4.0*". Every electrode placement site is previously clean using cotton with alcohol in an effort to reduce electrode-skin impedance, and fixed to the skin with an EEG proper paste "*ELEFIX*".

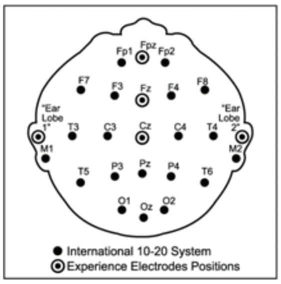


Figure 3. International 10-20 System (Electrodes' Position)

Every electrode is placed according to the international 10-20 System (Fig. 3). The montage used is Cz (Channel 1) and Fz (Channel 2) as the active electrodes, left ear lobe (Channel 1) and right ear lobe (Channel 2) as the reference ones, and Fpz as the ground electrode.
Sampling frequency is set at 1000 samples per second (1000Hz) and 15 stimuli are used per class in a shuffle way. Whenever the individual identifies the visual or auditory stimulus, previously selected, he or she triggers a button on "*RB-730 Response pad by Cedrus Corporation*" that registers on "*Acqknowledge*" his recognition and, subsequently, the EEG Evoked Potential signal. Visual and auditory stimuli are presented, respectively, with 1,5s and 2,0s of duration. The first ones through a display, and the second ones through earphones. During the acquisition time the volunteer was seated, keeping calm and relaxed in a dark and silent room.

All signals are processed and averaged techniques are applied to it using "*MatLab*" software. We used "*MatLab*" software to develop the algorithm due to the fact that it is a more flexible tool and it allows us a deeper degree of analysis.

4. Results: Work in progress

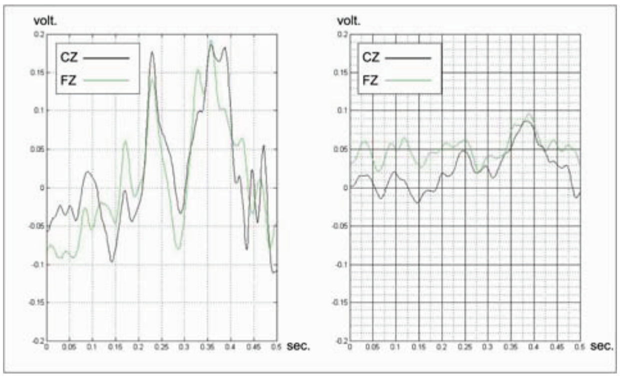


Figure 4. Event Related Potential with both channels overlapped: Pure tone at 4KHz

Insofar as in this poster (short paper) we want to demonstrate our process and methodology regarding to the Event Related Potentials working process study, our team reserve to ulterior time all the results regarding to this research. The image above (Fig. 4) explicit for now as a valid and concrete example an Event Related Potential measurement withdrawn of one set 4 KHz stimulus. Here we can analyze on 500 milliseconds latency window the energy generated per time, waveform, the presence or absence of N2 (or N200) – a negative peak (upward direction) that appears at approximately 200 ms - and P3 (or P300) – a positive peak (downward direction) that appears around 300 ms [Hruby; Marsalek 2003], the amplitude and latency of the waves and, with that, systematize and correlate the different stimuli. So now, we will prosecute the measurements of all samples collected with the same procedures and methodology here described.

6. References

Arnheim, Rudolf "Art and Visual Perception: A Psychology of the Creative Eye". The New Version. California: University of California Press, ISBN 0-520-02613-6

Blum, A. S., Rutkove, S. B. "The Clinical Neurophysiology Primer". illustrated edition. Humana Press, 2007. pp. 46; 461-482. eISBN 978-1-59745-271-7; ISBN 0-89603-996-X (alk. paper).

Chiappa, K. H. "Evoked potentials in clinical medicine". 3rd edition, illustrated. Lippincott Williams & Wilkins, 1997. pp. 1-2; ISBN 157-158. 0397516592, 9780397516599.

Dondis, D. A.- "La Sintaxis de la Imagen". 6th. Barcelona: Gustavo Gili, 1985. ISBN 84-252-0609-X

Hubry, T., Marsalek, P. "Event Related Potentials – the P3 Wave". Acta Neurobiol. Exp. 2003, 63: 55-63

Niedermeyer, E., Lopes da Silva, F. "Electroencephalography: basic principles, clinical applications, and related fields". 5th edition, ilustrated. Lippincott Williams & Wilkins, 2004. pp. 1017- 1027; 1045-1047. ISBN 0781751268, 9780781751261.

Walsh, P., Kane, N., Butler, S. "The Clinical Role of Evoked Potentials". Journal of Neurology, Neurosurgery and Psychiatry. 76, 2005, pp. (Suppl II):ii16–ii22.

Goiaba: a software to process musical contours

Marcos S. Sampaio[1], **Pedro Kröger**[1]

[1]Genos—Grupo de pesquisa em Computação Musical
Escola de Música da Universidade Federal da Bahia

mdsmus@gmail.com, pedro.kroger@gmail.com

Abstract. *Contour is the shape or format of objects. Contours can be associated to musical parameters such as pitch and time, representing one in function of another. Contours help to give coherence to musical piece and can be used to analyze and to compose music. Contour theories provide many operations that demand precise mathematical calculation. In this article we present the current state of Goiaba, a software that assists musicians in contour related tasks such as the calculation and plotting of operations, and a case study of a composition where Goiaba was used to generate the contour-related material.*

1. Introduction

Contour is the shape or format of an object. In music one can speak of a pitch contour, density contour, and so on. Contours can easily be recognized from graphic representation by professionals and laymen alike [Marvin, 1988]. For instance, Beethoven's Fifth Symphony's main motive and the corresponding pitch contour are represented respectively in figures 1a and 1b.

(a) Main motive

(b) Contour F(3 1 2 0)

Figure 1: Fifth Symphony main motive contour

Technically, a contour is an ordered set of numbered elements [Morris, 1993]. Absolute values of contour elements are ignored, and only the high-low relationship between them is regarded. For instance, the music in figures 1a and 2 have the same pitch contour, graphically represented in figure 1b, and symbolically by F(3 1 2 0)[1]. Yet, both passages sound completely different. In our opinion that is a feature of using contour theory in composition, to have an underlining process providing coherence and musical variety at the same time.

The study of contour is important because contours can help to give coherence to a musical piece, like motives and pitch sets. They are structural devices that can be combined through operations like inversion and retrogradation, and can be approached by analytical or compositional points of view.

[1]The uppercase letter F names the contour.

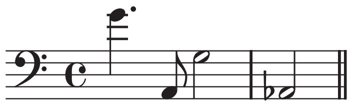


Figure 2: A melody with the F(3 1 2 0) contour

Contour theories [Friedmann, 1985, Morris, 1987, Marvin, 1988, Beard, 2003] have been developed to organize the current knowledge about contour in a systematic way. These theories were developed primarily as analytic techniques for non-tonal compositions [Beard, 2003], and provide arrays, matrices and many operations to help the comparison of contours, like inversion, translation, comparison matrix, and contour interval array. It's out of the scope of this paper to provide a comprehensive review about contour theory. The reader can find a good literature review in [Beard, 2003].

A computer program to process contours can assist composers and analysts in tasks like the calculation of operations—avoiding human error and wasting time—, automated graphical plotting, and conversion from music score to contours and vice-versa. For these reasons we are developing *Goiaba*, a software to contour processing (described in section 2).

In this article we will present *Goiaba*, its data representation, examples of the software output, and a case study of a composition, in which *Goiaba* was used to compose most of a woodwind quintet.

2. The contour processing program Goiaba

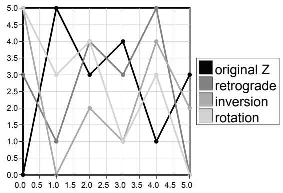
Goiaba is a program written in Common Lisp developed by the authors of this paper to process and plot contours. It has many contour-related operations, like inversion, retrogradation, rotation, contour reduction [Adams, 1976], contour class, contour adjacency series, contour adjacency series vector, contour interval, contour interval array, contour class vector I and II [Friedmann, 1985], and comparison matrix [Morris, 1993]. Currently, *Goiaba* accepts and shows contours in a numeric format, but it can also plot contours in a pdf file.

Goiaba has two representations for contours; simple contours represents only the values of the contour elements, and contours with durations are basically ordered collections of cartesian points. For instance, the contour in figure 1b would be represented as a simple contour as `#s(3 1 2 0)` and as a contour with duration as `#d(#p(0 3) #p(1 1) #p(2 2) #p(3 0))`. The forms `#s(...)` and `#d(...)` indicate a simple contour and a contour with duration, respectively. The notation `#p(x y)` indicates a point with two values. So, from the example we can see that *Goiaba* really represents a contour with duration as a collection of points. The symbols #s, #d, and #p are user-defined lisp reader macros that expand into code to instantiate objects of types simple-contour, contour-with-duration, and point, respectively. For instance, `#p(0 1)` is expanded to `(make-instance 'point :x 0 :y 1)`, which is the usual way of instantiating objects in common lisp. Finally, *Goiaba* has a few constructor functions besides the reader macros to help the creation of contour objects. The functions, `make-point-list`, `make-simple-contourlist`, and `make-contour-with-duration-list`, map a list to the correspondent object.

Goiaba uses the Cl-pdf library to plot contours, allowing easy visualization of contour operations. For instance, the code in figure 3a generates a graph with the original contour Z, `#s(0 5 3 4 1 3)`, and its retrogradation, inversion, and rotation. The result can be seen in figure 3b.

```
(plot "contour.pdf"
  Z "original Z" :red
  (retrograde Z) "retrograde" :blue
  (inversion Z) "inversion" :orange
  (rotation Z 1) "rotation" :green)
```

(a) Code to compute and plot a few operations



(b) Output: Z(0 5 3 4 1 3) contour operations

Figure 3: *Goiaba* input code and graphic output

Goiaba takes advantages of Common Lisp's multimethods capabilities. In Common Lisp a method is actually a generic function that specializes on the type of its arguments. For example, we have two definitions for a generic function called `rotate`, one specializes on the contour-with-duration object while the other on the simple-contour object. The advantage of this approach is that we can add more types of contour if necessary and write the appropriate generic functions that will specialize on those types, without disrupting the existing code.

3. The application of Contour in composition

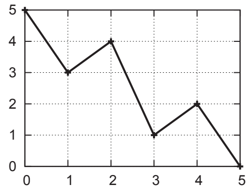
Systematic studies about the usage of contour operations and combinations in musical composition are scarce, despite the possible coherence that contours can provide. For this reason we are researching the usage of contour in composition and its advantages. The first author of this paper, during his master's [Sampaio, 2008], composed a woodwind quintet, based on contour theories operations. We will use this piece as a case study for the use of contour theory in composition, and how *Goiaba* was useful in the compositional process.

This piece was composed entirely using *Goiaba* to simplify the calculation of contour operations and plotting. The piece is based on the contour P(5 3 4 1 2 0) and on combinations of contour operations associated to parameters such as pitch, tempo, density and texture. In figure 4a we can see the original contour P and its subsets and operations; retrogradation, inversion, rotation, and interpolation.

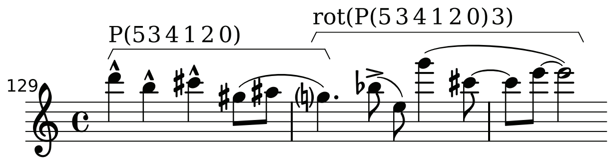
Goiaba was essential to compose a *fugato* session in the quintet because each part of the subject and countersubject were based on different combinations of operations of rotation and retrogradation. The subject is formed by the concatenation of P and its rotation by a factor of 3, as seen in figure 4b. Figure 5a shows *Goiaba*'s graphical output for both contours. The countersubject is formed by a sequence of three rotations (by the factor of 5, 4, and 3) of the retrograde of P (fig. 4c). We can see the graphical output of these contours in figure 5b.

4. Conclusion and future work

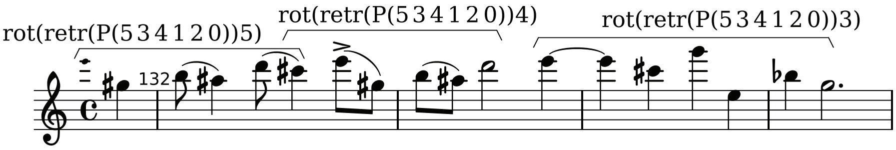
Contours help to give coherence to a musical piece, are easily recognized graphically by musicians and laymen, and can be used to analyze and compose music. Contour theories provide many operations that demand precise mathematical calculations and graphical representation, for this reason we are developing *Goiaba*, a contour processing software that helps the calculation and plotting of contour operations. *Goiaba* has been proven to be useful in composing music that uses contour theory extensively. Currently *Goiaba*



(a) The P(5 3 4 1 2 0) contour

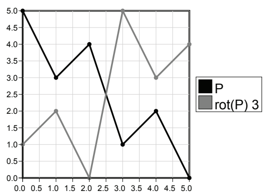


(b) Subject

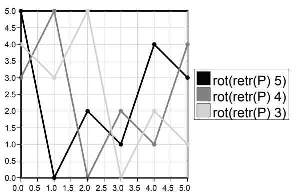


(c) Counter-subject

Figure 4: Structural elements of *fugato*



(a) Subject



(b) Counter-subject

Figure 5: Software output for *fugato* contour operations

accepts only input in a symbolic format, but we have plans to add support for Lilypond, MIDI, ABC, and MusicXML formats as well. The next step in our research is to improve *Goiaba* user interaction, releasing a more friendly interface, possibly with a GUI.

References

Adams, C. R. (1976). Melodic contour typology. *Ethnomusicology*, 20(2):179–215.

Beard, R. D. (2003). *Contour Modeling by Multiple Linear Regression of the Nineteen Piano Sonatas by Mozart*. Phd diss., The Florida State University. School of Music.

Friedmann, M. L. (1985). A methodology for the discussion of contour: its application to schoenberg's music. *Journal of Music Theory*, 29(2):223–48.

Marvin, E. W. (1988). *A generalized theory of musical contour: its application to melodic and rhythmic analysis of non-tonal music and its perceptual and pedagogical implications*. PhD thesis, University of Rochester.

Morris, R. D. (1987). *Composition with Pitch-classes: A Theory of Compositional Design*. Yale University Press.

Morris, R. D. (1993). New directions in the theory and analysis of musical contour. *Music Theory Spectrum*, xv:205–28.

Sampaio, M. (2008). Em torno da romã: aplicações de operações de contornos na composição. Master's thesis, Universidade Federal da Bahia, Salvador, BA.

Functional Harmonic Analysis and Computational Musicology in Rameau

Alexandre Tachard Passos[1], Marcos Sampaio[2], Pedro Kröger[2], Givaldo de Cidra[2]

[1]Instituto de Computação (IC)
Universidade Estadual de Campinas (UNICAMP)
Campinas, Brazil

[2]Genos—Computer Music Research Group
School of Music (EMUS)
Federal University of Bahia (UFBA)
Salvador, Brazil

Abstract. *In this paper we present the infrastructure for computational musicology and functional harmonic analysis in Rameau, a framework for experimentation with musicological ideas in software. Rameau supports out of the box chord labeling, key finding, tonal function detection, cadence detection, voice crossing identification, parallel fifths and octaves recognition, seventh note resolution analysis, and can be easily extended to support many other features. It can also generate textual reports, graphical visualization, and typeset scores with the results of these analyses. Rameau is fully open source and implemented in Common Lisp.*

1. Introduction

In the past few years we have been developing Rameau, an open-source system for automatic harmonic analysis and computational musicology [Kröger et al. 2008a]. With Rameau one can process Lilypond files to generate typeset harmonically analysed scores, identify non-chordal sonorities and label chords, perform some basic musicological tasks (such as cadence and voice crossing detection). Rameau is written in Common Lisp and its source code can be found in `http://genos.mus.br/rameau/`, together with the data set of Bach chorales from the Riemenschneider [Riemenschneider 1941] edition we use to benchmark and test the system. In this paper we present Rameau's infrastructure for computational musicology and Rameau's implementation of a functional harmonic analysis framework.

The organization of this paper is as follows; section 2 gives an overview of how Rameau works, section 3 presents the main musicological features implemented in the system. Section 4 describes the functional harmonic analysis structure of Rameau. Section 5 describes superficially each algorithm currently implemented in Rameau. Section 6 contains some concluding remarks and directions for future work.

2. Overview of Rameau

There are two ways of accessing Rameau: a comprehensive command-line interface and a simple web server to perform functional harmonic analysis. When starting the command-line interface the user can choose which Lilypond files are to be processed and what to do with them. The possible operations are called *commands*, and some of the available commands are:

- **help**: provides help and describes the other commands individually;
- **octaves**: identifies parallel octaves in the specified files;
- **crossings**: detects voice crossings in the specified bach chorales;
- **cadences**: detects the cadences in the specified files;
- **analysis**: identifies non-chordal sonorities and name the chords in the specified files, with many different algorithms;
- **functional**: performs functional harmomnic analysis of the given files;
- **document**: generates html documentation for the Rameau source code.

There are altogether 26 different commands, and a new useful command can be implemented in five lines of common lisp code.

Musicological commands can output their data in many formats, such as a typeset score of the interesting section of the song, a list of interesting matches, a histogram of interesting matches and a cloud plot of the extracted data. The harmonic analysis commands (**functional** and **analysis**) can generate tabular results and annotated typeset scores.

3. Computational musicology

The commands **octaves** and **fifths** show how many consecutive octaves and fifths are in a piece and where they are. We found that all consecutive octaves in Bach Chorales are in the form unison–octave or octave–unison, but no consecutive octaves in all Chorales are parallel, although a few fifths are (in chorales 4, 46, 71, and 266). More information can be found in [Kröger et al. 2008b].

The command **chords** lists the frequency of each type of chord in a set of chorales. The command **crossings** finds passages where are voice crossings. We found that in 57% of the Bach chorales there is some kind of voice crossing, although most of the crossings happen in a short period of time (no more than two beats). There are a few interesting cases. For instance, the alto is the lowest voice for a brief period of time in chorale #35 and there is a crossing of the soprano and alto and tenor and bass at the same time in chorale # 290.

There are also commands to find the vocal range used in a composition (**kostka-amb**), to find melodic jumps in a voice (**jumps**), to collect statistics on seventh notes' resolution (**resolve-seventh**), to collect data on how many chord progressions found in the chorales are strong, weak, superstrong and neutral, according to Schoenberg's theory of harmony [Schoenberg 1983] (**schoenberg**), to detect the final cadences in the analyzed pieces (**cadences**), and to count the notes found in major and minor modes (**count-major-notes** and **count-minor-notes**).

4. The functional harmonic analysis structure of Rameau

Roman numeral functional analysis consists, roughly, of two activities: key finding—determining what is the tonal center of the piece and its parts—and roman numeral function detection—determining the tonal function of each segment of the piece. In Rameau we chose to merge these two conceptual concerns into one, and the internal representation chosen reflects that, by stating that the analysis of a song is a list of the analyses of every distinct sonority in the song, and the analysis for each sonority is a local key, mode, and tonal function.

One of the goals of the Rameau project to understand and compare previously published and new techniques for automatic harmonic analysis. To properly compare harmonic analysis algorithms we have a corpus of 371 Bach chorales from the Riemenschneider edition [Riemenschneider 1941], 20 of them annotated by experts with an acceptable harmonic analysis. Rameau includes facilities to train machine learning algorithms on these analyzed chorales (commands named **algorithms** and **funalg** for chord-labeling and functional analysis algorithms, respectively) and a bayesian framework for estimating the correctness of the expert annotations and using this estimate to derive confidence intervals for the accuracies of the algorithms (command named **information-theory**). The analysis results are automatically typeset with the aid of the Lilypond music typesetting program [Nienhuys and Nieuwenhuizen 2008].

5. Functional harmonic analysis algorithms

Using the infrastructure described in section 4 Rameau currently has implementations of four different roman numeral functional analysis; a hidden Markov model, a k-nearest neighbors classifier, a neural network-based harmonic analyzer similar to the one described by Tsui [Tsui 2002], and a trivial extension of Pardo & Birmingham's chord labeling algorithm [Pardo and Birmingham 1999].

5.1. Hidden Markov Model

A hidden Markov model is any probabilistic function of a Markov chain. Using a hidden Markov model to perform roman numeral function analysis consists of modeling the notes in a tonal piece as a probabilistic function of the underlying harmonies, and finding these harmonies, given the notes, using traditional hidden Markov model algorithms. Our approach closely follows that of Raphael and Stoddard [Raphael and Stoddard 2003], and the differences are noted in [Passos 2008].

5.2. K-Nearest Neighbors

A tool used in machine learning for many non-trivial tasks is the k-nearest neighbors classifier [Mitchell 1997]. It works by first representing the instances to be classified in some metric space. Then, to classify an instance x, the knn algorithm chooses, from the training data set, the set s of the k closest examples to x and outputs the most common class in s. The spatial representation we chose for Rameau is a pitch frequency array a, in which, if $f * n$ on the n pitches in a given sonority are encoded as having number p, then $a[p] = f$. When considering surrounding context, we concatenate these arrays and, to avoid adding too much noise in the distance function we weight them down in proportion to the square of the distance between the contextual sonority and the sonority being analyzed. More details can be found in [Passos 2008].

5.3. Pardo & Birmingham's

Pardo & Birmingham [Pardo and Birmingham 1999] describes an algorithm for chord labeling that has some predefined chord templates and chooses among them the one that most closely matches the notes sounding in a given sonority. To extend the original algorithm to perform roman numeral functional analysis we simply created the key for the whole piece using the root and mode of the first chord found, and thus computed the roman numeral function for all other chords as if they were in that key. While simplistic, this approach performs almost competitively with the hidden Markov model.

6. Conclusions and future work

In this paper we presented the current status of Rameau, a framework for automatic harmonic analysis and computational musicology. The framework is mature, and has implementations of many useful musicological functions. The architecture is still too tied to 4-voice part writing, command-line operation, and the Lilypond format, but we are working to correct this in future releases.

While still preliminary, the current implementation of functional harmonic analysis in Rameau is promising, and already produces useful results. Rameau has implementations of a hidden Markov model, a K-nearest neighbors, Pardo & Birmingham's, and neural networks functional harmonic analysis algorithms. Rameau is open source, written in Common Lisp and its source code (together with our data sets and instructions on how to compile, install and run it) is available at `http://genos.mus.br/rameau/`.

References

[Kröger et al. 2008a] Kröger, P., Passos, A., Sampaio, M., and de Cidra, G. (2008a). Rameau: A system for automatic harmonic analysis. In *Proceedings of the 2008 International Computer Music Conference*, pages 273–281.

[Kröger et al. 2008b] Kröger, P., Passos, A., Sampaio, M., de Cidra, G., de Souza Ourives, N., dos Anjos, E. S., and dos Santos, W. S. (2008b). Musicologia computacional aplicada à análise dos corais de bach. In *Anais do XVIII Congresso da ANPPOM*.

[Mitchell 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

[Nienhuys and Nieuwenhuizen 2008] Nienhuys, H.-W. and Nieuwenhuizen, J. (2008). Lilypond. Available at `www.lilypond.org`.

[Pardo and Birmingham 1999] Pardo, B. and Birmingham, W. P. (1999). Automated partitioning of tonal music. Technical report, Electrical Engineering and Computer Science Department, University of Michigan.

[Passos 2008] Passos, A. (2008). Modelos de markov ocultos para análise harmônica funcional de música tonal. Monografia de Graduação defendida na Universidade Federal da Bahia.

[Raphael and Stoddard 2003] Raphael, C. and Stoddard, J. (2003). Harmonic analysis with probabilistic graphical models. In Hoos, H. H. and Bainbridge, D., editors, *Proceedings of ISMIR 2003*, Baltimore, Maryland, USA. The Johns Hopkins University.

[Riemenschneider 1941] Riemenschneider, A., editor (1941). *371 Harmonized Chorales and 69 Chorale Melodies with figured bass*. G. Schirmer, New York.

[Schoenberg 1983] Schoenberg, A. (1983). *Theory of Harmony*. University of California Press.

[Tsui 2002] Tsui, W. S. V. (2002). Harmonic analysis using neural networks. Master's thesis, University of Toronto.

Reviewers

Adina Izarra	Jonatas Manzolli
Adolfo Maia Jr.	Joseph Thibodeau
Alejandro Viñao	Luciano Flores
Aluizio Arcela	Luis Jure
Alvaro Barbosa	Marcelo Johann
Andrew Horner	Marcelo Pimenta
Catalina Peralta Cáceres	Marcelo Queiroz
Chris Chafe	Marcelo Wanderley
Damián Keller	Marcio Brandão
Daniel Teruggi	Marisa Pedemonte
Eduardo Miranda	Mikhail Malt
Emilios Cambouropoulous	Nelson Lago
Evandro Miletto	Oscar Di Liscia
Fabien Gouyon	Peter Beyls
Fabio Kon	Petri Toiviainen
Fernando Iazzetta	Regis R. A. Faria
Fernando Lopez-Lezcano	Rob Hamilton
Flávio Soares Corrêa da Silva	Rodolfo Coelho de Souza
Geber Ramalho	Roger Dannenberg
Gianpaolo Evangelista	Rosa Viccari
Giordano Cabral	Sever Tipei
Hugo de Paula	Stephen Sinclair
Ian Whalley	Vanessa Yaremchuk
Jean-Pierre Briot	Victor Lazzarini

Author Index

ISSN 2175-6759