Proceedings

UFSJ

17º SBCM

2019

17º Brazilian Symposium on Computer Music

Federal University of São João del-Rei

25-27 September 2019

http://compmus.ime.usp.br/sbcm/2019/

17th Brazilian Symposium on Computer Music

Federal University of São João del-Rei
25-27 September 2019
São João del-Rei - MG - Brazil

Proceedings

ISBN - 978-85-7669-490-8

ISSN - 2175-6759

Edited by
João Teixeira Araújo
Flávio Luiz Schiavoni

Organized by
Flávio Schiavoni, Federal University of São João del Rei
Tiago Tavares, State University of Campinas
Rogério Constante Tavares, Federal University of Pelotas
Régis Rossi Alves Faria, University of São Paulo

Cover design
Fábio dos Passos Carvalho

Ficha catalográfica elaborada pelo Setor de Processamento Técnico da Divisão de Biblioteca da UFSJ

Brazilian Symposium on Computer Music (17. : 2019 : São João del-Rei)

Proceedings [recurso eletrônico] do 17º Brazilian Symposium on Computer Music (SBCM), 25 a 27 de setembro de 2019, São João del-Rei, MG / editado por João Teixeira Araújo e Flávio Luiz Schiavoni. – São João del-Rei: UFSJ, 2019.

Disponível em: http://compmus.ime.usp.br/sbcm/2019/assets/proceedings.pdf
ISSN: 2175-6759
ISBN: 978-85-7669-490-8

1. Música por computador. 2. Arte e tecnologia. 3. Multimídia (Arte). I. Araújo, João Teixeira (Ed.). II. Schiavoni, Flávio Luiz (Ed.). III. Título.

CDU: 78:004

CNPJ nº 29.532.264/0001-78
http://www.sbc.org.br
Av. Bento Gonçalves, 9500 – Setor 4 – Sala 116 – Prédio 43424 – Agronomia
CEP 91501-970 – Porto Alegre – RS, Brasil

SBCM 2019 is organized by
Arts Lab in Interfaces, Computers, and Everything Else (ALICE / UFSJ)
Computer Science Department (DCOMP / UFSJ)
Federal University of São João del-Rei (UFSJ)

and has the support and promotion of the following institutions

Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)
Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)
Brazilian Computing Society (SBC)
Pró-Reitoria de Pesquisa e Pós Graduação (PROPE / UFSJ)
Pró-Reitoria de Extensão e Cultura (PROEX / UFSJ)
Departamento de Computação (DCOMP / UFSJ)
Programa de Pós Graduação em Ciência da Computação (PPGCC / UFSJ)
Programa Interdepartamental de Pós-graduação Interdisciplinar em Artes, Urbanidades e Sustentabilidade (PIPAUS / UFSJ)

Sponsors

Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)
Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)
Pró-Reitoria de Pesquisa e Pós Graduação (PROPE / UFSJ)

Acknowledgements

Introduction

Welcome to the 17th edition of SBCM! The Brazilian Symposia on Computer Music are thriving and exciting venues for sharing ideas about recent developments in the fields of computer music, sound and music processing, music information retrieval, computational musicology, multimedia performance and many other things related to art, science and technology. The event takes place at the Federal University of São João del-Rei, Brazil, from September 25th to September 27th 2019, and has as Keynote Speakers Dra. Isabel Cecilia Martínez (Directress of the Laboratory for the Study of the Musical Experience at the Universidad Nacional de La Plata in Argentina), Martín Rocamora (member of the IEEE - Institute of Electrical and Electronics Engineers, the AES - Audio Engineering Society and the SNI - Sistema Nacional de Investigadores in Uruguay), Maurício Alves Loureiro (Coordinator of the research group CEGeME - Center for Studies of Musical Gesture and Expression and directed the IEAT - Institute of Advanced Transdisciplinary Studies of UFMG) and Fernando Iazzetta (consultant for the Arts Committee at FAPESP – the São Paulo Research Foundation).

The program of the conference includes keynote talks, oral presentations of music and technical papers, poster discussion sessions, discussion panels and concerts, providing plenty of opportunities for interaction and discussion as a way to foster collaborations and novel ideas for the critical problems of our related fields. The present volume features the contributions presented at SBCM 2019, including full technical papers, full music papers, posters and art, which express the ongoing exchange taking place among the fields of music, computer science and engineering, among others, and their contributions to the advancement of scientific and artistic practices.

Since 1994, the Brazilian Computer Music Symposia have provided a window into the state-of-the-art developments in this rich intersection of ideas, interests and competences which converge into interdisciplinary work. The 2019 edition emphasizes current research in audio open-software resources, ubiquitous music, music information retrieval and music cognition, featuring contributions from Latin America and other research networks around the world. In this edition, the call for art, music and scientific contributions received 50 submissions of full papers, 12 poster submissions, and 24 art submissions. The double-blind, peer-reviewed process involved 45 reviewers, contributing with almost 260 evaluations that lead to an acceptance rate of 45% for full technical papers and 57% overall. Such an outstanding response of the participant interdisciplinary research communities is represented in the technical and artistic program, being complemented with the full versions of the selected submissions in these proceedings.

We hope you will benefit from it!

The SBCM 2019 Organizing committee

Invited talks

Music, Embodied Mind, and Cultural Practice: how the self and the other shape musical experience

Dra. Isabel Cecilia Martínez
Laboratorio para el Estudio de la Experiencia Musical.
Facultad de Bellas Artes.
Universidad Nacional de La Plata. Argentina
http://leem.fba.unlp.edu.ar/el-laboratorio/integrantes/dra-isabel-cecilia-martinez/

In this talk, human musicality is inquired in the context of the cultural practice of music. Informed by the theory of social embodied cognition, some musical practices are investigated with the aim of accounting for an ontology of the self and the other in the cultural practice of music. Since the very beginning of life, making sense of music requires human action-perception involvement with the complexity of sonic-kinetic events. This capacity is based on human disposition to build social meaning throughout the interaction of our mind-body-environmental complexes with others'. Musical development is bound to the ways in which the temporal, spatial and dynamic configurations of sound and movement are organized in our self-other interaction, and developed in our imaginative, embodied and emotional cognition. The meaning of the embodied mind and the ways the self and the other in interaction shape human experience in the cultural practice of music are important to discuss an epistemological turn in the fields of musicology, music psychology and music pedagogy, and also to open new music research avenues.

About Isabel:

Doctora en Psicología de la Música por la Universidad de Roehampton Surrey, Reino Unido. Es Licenciada y Profesora en Educación Musical egresada de la Universidad Nacional de La Plata. Es Profesora Titular de las cátedras de Metodología de las Asignaturas Profesionales y Audioperceptiva 1 y 2 en la Facultad de Bellas Artes de la UNLP. Es Docente Investigadora Categoría I y Directora del Laboratorio para el Estudio de la Experiencia Musical (LEEM-FBA-UNLP). Dirige un equipo de investigadores, becarios y tesistas de la UNLP en el proyecto de investigación "La corporeidad de la mente musical. Hacia una definición de su estatura en el estudio de la ontogénesis, la percepción y la performance de la música". Es directora del Proyecto PICT 2013-0368 "Musicalidad Comunicativa en las Artes Temporales y la Temprana Infancia" subsidiado por la Agencia para la Investigación Científica y Tecnológica (FONCYT). Es editora de la revista Epistemus y miembro del comité editorial de varias publicaciones académicas internacionales. Ha dictado cursos y conferencias en varios países de Latinoamérica y Europa y es profesora en diversos posgrados de universidades argentinas y extranjeras. Es miembro fundador y Presidente actual de la Sociedad Argentina para las Ciencias Cognitivas de la Música (SACCoM). Investiga aspectos de la cognición musical corporeizada y el pensamiento metafórico en la música y sus implicancias para la teoría y la práctica de la enseñanza en la formación musical. Ha publicado y difundido su investigación en el ámbito nacional e internacional.

The First Brazilian Symposium on Computer Music presents Brazilian computer music potentials - Caxambu, MG, 1994

Dr. Maurício Alves Loureiro
CEGeME - Center for Studies of Musical Gesture and Expression
IEAT - Institute of Advanced Transdisciplinary Studies of UFMG
Federal University of Minas Gerais
http://musica.ufmg.br/cegeme/

The main objective of this talk is to report on the First Brazilian Symposium on Computer Music, which occurred on August 1994, at the city of Caxambu, Minas Gerais, promoted by the UFMG. NUCOM, the group of young academic dedicated to this emerging research field in Brazil, was created as a discussion list by e-mail, during the year of 1993. This quite exciting and fancy event at Hotel Gloria in Caxambu was able to imposingly launch the group to the national, as well as to the international academic community. First, due to the excellency of the event's output and its daring program, that included 34 selected papers by researchers from various institutions from Argentina, Brazil, Canada, Denmark, France, Hong Kong, Mexico, UK, and USA, 5 lectures an 2 panels of discussion offered by researchers from the most important computer music research centers all over the world. The program also included eight concerts, two of them featuring traditional music, such as Bach, Mozart, and Brazilian music. Six computer music concerts presented 48 selected compositions submitted to the symposium. Second, as the symposium happened as apart of the 14th Congress of Brazilian Computer Science Society (SBC), the excellency of its output was able to attract the interest of SBC's board of directors. They invited NUCOM to integrate the society as Special Committee, which are sub-groups of SBC dedicated to specific computer science topics. At the end of the description, this report aims at raising questions, arguments, and debates about today's format of NUCOM meetings, considering more seriously the interdisciplinary character of the methodologic approaches adopted by the field. Interdisciplinarity should be pursued by striving to contaminate a growing number of different topics of musical sciences, as well as of other research fields.

About Maurício:

Aeronautical Engineer, graduated at the Technological Institute of Aeronautics – ITA (1976) and clarinetist, graduated at the Staatliche Hochschule für Musik Freiburg, Germany (1983), sponsored by DAAD, where he studied with famous clarinetist Dieter Klöcker. In 1991 he obtained a Doctor degree in Music at the University of Iowa, USA, where he also studied electronic and computer music. In 1985 he was appointed as the first clarinet assistant of the State Symphony Orchestra of São Paulo, the most prominent Brazilian Symphony Orchestra. He has appeared as soloist with some of the leading orchestras in Brazil and participated as invited artist on major music festivals in Brazil and in USA. In 1994, he organized the first Brazilian Symposium for Computer Music, as well as its fifth (1998) and tenth versions (2005). He was associated professor at the Institute of Arts of the State University of São Paulo - UNESP (1984-1992) and is now full professor at the School of Music of the Federal University of Minas Gerais - UFMG, where he coordinates the research group CEGeME - Center for Studies of Musical Gesture and Expression and directed the IEAT - Institute of Advanced Transdisciplinary Studies of UFMG (2009-2013).

The Politics of Computer Music

Dr. Fernando Henrique de Oliveira Iazzetta
NuSom – Núcleo de Pesquisas em Sonologia
University of São Paulo
http://www2.eca.usp.br/nusom/

When a set of objects, actions, and procedures begin to coalesce and gain some coherence, they become perceived as a new, cohesive field. This may be related to the emergence of a new discipline, a new craft, or a new technological configuration. As this new field shows some coherence and unity, we tend to overlook the conditions that gave rise to it. These conditions become "naturalized" as if they were inherent in that field. From this point on, we do not wonder anymore to what extent the contingencies (formal, social, economic, technological, aesthetic, religious) that gave rise to that field have been crucial to its constitution.

When it comes to computer music we are used to its applied perspective: tools, logical models, and algorithms are created to solve problems without questioning the (non-computational) origin of these problems or the directions taken by the solutions we give to them. The idea of computing as a set of abstract machines often hides the various aspects of the sonic cultures that are at play when we develop tools and models in computer music.

The way we connect the development of computer tools with the contingencies and contexts in which these tools are used is what I call the politics of computer music. This connection is often overshadowed in the development of computer music. However, I would like to argue that this connection is behind everything we do in terms of computer music to the point that it often guides the research, development, and results within the field. In this presentation, I would like to consider the politics of computer music as a way to critically explore the field. I'll also point to some initiatives in this direction that we have developed at the NuSom, the Research Center on Sonology of the University of São Paulo.

About Fernando:

Fernando Iazzetta is a Brazilian composer and performer. He teaches music technology and electroacoustic composition at the University of São Paulo and is the director of NuSom – Research Centre on Sonology at the same university. His works have been presented in concerts and music festivals in Brazil and abroad. As a researcher he has been interested in the investigation of experimental forms of music and sound art. He also runs a record label and studio – the LAMI – at the University of São Paulo. He currently is a research fellow at CNPq, the Brazilian National Council of Scientific and Technological Development. Since 2010 he is the consultant for the Arts Committee at FAPESP – the São Paulo Research Foundation.

Computational Methods for Percussion Music Analysis

Dr. Martín Rocamora
Universidad de la República (UdelaR), Uruguay
IEEE - Institute of Electrical and Electronics Engineers
AES - Audio Engineering Society
SNI - Sistema Nacional de Investigadores in Uruguay
https://iie.fing.edu.uy/~rocamora/

Most of the research conducted on information technologies applied to music has been oriented towards mainstream popular music in the so-called 'Western' tradition. Although it proved to be effective for various music styles and repertoires, new approaches are needed to deal with other music traditions, such as those from Africa, China, India, or the Arab world. Fortunately, over the last few years there have been increasing efforts devoted to the study of traditional, folk or ethnic music. The computational analysis of rhythm from audio signals remains a challenging task in several cases, for instance, for syncopated or poly-rhythmic music.
This talk offers an overview of the research we conducted over the last few years on computational rhythm analysis from audio recordings, considering the Afro-Uruguayan candombe drumming as a case study. It comprises the creation of datasets, the discovery and analysis of rhythmic patterns, the study of micro-timing and the development of algorithms for beat and downbeat tracking. Besides, it also discusses our current efforts to improve and extend the methods to other music traditions, in particular, to Afro-Brazilian Samba.

About Martín:

Martín Rocamora is Assistant Professor in Signal Processing at Universidad de la República (UdelaR), Uruguay. He holds B.Sc, M.Sc., and D.Sc. degrees in Electrical Engineering from the School of Engineering (UdelaR). He was Teaching Assistant in Music Technology at the School of Music (UdelaR). His research focuses on the application of machine learning and signal processing to audio signals, with applications in machine listening, music information retrieval, and computational musicology. He is currently a member of the IEEE (Institute of Electrical and Electronics Engineers), the AES (Audio Engineering Society) and the SNI (Sistema Nacional de Investigadores) in Uruguay. Please visit his personal website for a complete list of publications and code/data releases (https://iie.fing.edu.uy/~rocamora/).

Committees

Organizing Committee

Flávio Schiavoni, Federal University of São João del Rei (General Chair)
Tiago Tavares, State University of Campinas (Technical Papers Chair)
Rogério Constante Tavares, Federal University of Pelotas (Music Chair)
Régis Rossi Alves Faria, University of São Paulo (Music Chair)
Rodrigo Schramm, Federal University of Rio Grande do Sul
Charles Figueredo de Barros, Federal University of São João del-Rei
Marcos Edson Cardoso Filho, Federal University of São João del-Rei
Edimilson Batista dos Santos, Federal University of São João del-Rei

Local Arrangements Committee

André Lucas Nascimento Gomes, Federal University of São João del-Rei
Avner Maximiliano de Paulo, Federal University of São João del-Rei
Carlos Eduardo Oliveira de Souza
Fábio dos Passos Carvalho, Federal University of São João del-Rei
Frederico Ribeiro Resende, Federal University of São João del-Rei
Gabriel Lopes Rocha, Federal University of São João del-Rei
Igino de Oliveira Silva Júnior, Federal University of São João del-Rei
Igor Alves Estevam Andrade, Federal University of São João del-Rei
João Teixeira Araújo, Federal University of São João del-Rei
Jônatas Araújo da Silva, Federal University of São João del-Rei
Luísa Lugon-Moulin Brazil Santos, Federal University of São João del-Rei
Mauro César Fachina Canito, Federal University of São João del-Rei
Rafael Alves Soares de Andrade, Federal University of São João del-Rei
Rômulo Augusto Vieira Costa, Federal University of São João del-Rei
Thiago de Andrade Morandi, Federal University of São João del-Rei

Scientific/Artistic Committee

Aluizio Oliveira, Federal University of Minas Gerais (UFMG)
Andrew Horner, The Hong Kong University of Science & Technology
Charles Figueredo de Barros, Federal University of São João del Rei (UFSJ)
Clayton Mamedes, Federal University of Paraná
David Borgo, University of California San Diego (UCSD)
Diego Colombo Dias, Federal University of São João del-Rei (UFSJ)
Edilson Ferneda, Universidade Católica de Brasília - UCB
Edimilson Santos, Federal University of Sao Joao del-Rei

Eloi Fritsch, Federal University of Rio Grande do Sul (UFRGS)
Emilios Cambouropoulous, Aristotle University of Thessaloniki
Evandro Miletto, Instituto Federal do Rio Grande do Sul - Campus Porto Alegre
Fabio Kon, IME - University of São Paulo (USP)
Federico Visi, ICCMR, University of Plymouth
Felipe Tanios, University of Campinas
Fernando Cruz, Universidade de Brasília
Fernando Iazzetta, University of São Paulo (USP)
Flávio Schiavoni, Federal University of São João Del Rei
Isabel Nogueira, Federal University of Rio Grande do Sul (UFRGS)
Ivan Eiji Yamauchi Simurra, University of São Paulo (USP)
James Correa, Federal University of Pelotas
Jean Bresson, Institut de Recherche et Coordination Acoustique/Musique
Jean-Pierre Briot, LIP6 (CNRS - Univ. Paris 6) & PUC-Rio
Jose Fornari, State University of Campinas (UNICAMP)
Juliano Foleiss, Universidade Tecnológica Federal do Paraná
Leandro Costalonga, UFES
Luc Nijs, Ghent University
Luis Jure, Universidad de la República
Marcelo Pimenta, Federal University of Rio Grande do Sul (UFRGS)
Marcelo Queiroz, University of São Paulo (USP)
Marcos Laia, Federal University of São João Del Rei
Marcus Bittencourt, Universidade Estadual de Maringá - UEM
Martín Rocamora, Universidad de la República
Maurício Loureiro, Federal University of Minas Gerais (UFMG)
Mikhail Malt, Ircam-CNRS
Nuno Otero, Linnaeus University
Rael Toffolo, Estate University of Maringá
Regis Rossi A. Faria, University of São Paulo (USP)
Ricardo Bordini, Federal University of Bahia (UFBA)
Ricardo Dal Farra, Concordia University
Rodolfo Coelho de Souza, University of São Paulo (USP)
Rodrigo Schramm, Federal University of Rio Grande do Sul (UFRGS)
Roger Dannenberg, Carnegie Mellon University
Sever Tipei, University of Illinois School of Music
Tairone Magalhaes, Federal University of Minas Gerais (UFMG)
Tiago Tavares, Universidade Estadual de Campinas

Contents

Full Papers

A computer-based framework to analyze continuous and discontinuous textural works using psychoacoustics audio descriptors
Micael Antunes da Silva,Danilo Rossetti,Jonatas Manzolli 4

J-Analyzer: A Software for Computer-Assisted Analysis of Antônio Carlos Jobims Songs
Carlos Almada,João Penchel,Igor Chagas,Max Kuhn,Claudia Usai,Eduardo Cabral,Vinicius Braga,Ana Miccolis 12

Identifying Narrative Contexts in Brazilian Popular Music Lyrics Using Sparse Topic Models: A Comparison Between Human-Based and Machine-Based Classification
André Dalmora,Tiago Tavares 17

Iracema: a Python library for audio content analysis
Tairone Magalhaes,Felippe Barros,Maurício Loureiro 22

A technical approach of the audience participation in the performance ”O Chaos das 5“
João Teixeira,Avner Paulo,Igino Silva Junior,Flávio Schiavoni,Mauro César Fachina Canito,Rômulo Augusto Vieira Costa 28

BUZU: Retrieving public transportation data for sonification
Julian J. Arango,Esteban Viveros Astorga,Fernando Iazzetta,Cristiano Figueiró 35

Composing through Interaction: a framework for collaborative music composition based on human interaction on public spaces
Mauro Amazonas,Thais Castro,Rosiane de Freitas,Bruno Gadelha 42

State of art of real-time singing voice synthesis
Leonardo Brum,Edward David Moreno 50

Visualizing Air Drums: Analysis of Motion and Vocalization Data Related to Playing Imaginary Drums
Ana Julia Pereira Caetano,Tiago Tavares 58

Sustainable Interfaces for Music Expression
Igino Silva Junior,Flávio Schiavoni 63

Ha Dou Ken Music: Mapping a joysticks as a musical controller
Gabriel Lopes Rocha,João Teixeira,Flávio Schiavoni 69

TumTá and Pisada: Two Foot-controlled Digital Dance and Music Instruments Inspired by Popular Brazillian Traditions
João Tragtenberg,Filipe Calegario,Giordano Cabral,Geber Ramalho 76

Cognitive Offloading: Can ubimus technologies affect our musicality?
Leandro Costalonga,Marcelo Pimenta 84

Reflection and practice on generative music structuring: the formal problem of ergodicity
Gabriel e Silva,Clayton Mamedes 92

The development of libmosaic-sound: a library for sound design and an extension for the Mosaicode Programming Environment
Luan Gonçalves,Flávio Schiavoni 99

Combining Effects in a Music Programming Language based on Patterns
Andre Du Bois,Rodrigo Ribeiro 106

Prototyping Web instruments with Mosaicode
André Gomes,Frederico Resende,Luan Gonçalves,Flávio Schiavoni 114

PSYCHO library for Pure Data
Alexandre Porres 121

Low-Latency f0 Estimation for the Finger Plucked Electric Bass Guitar Using the Absolute Difference Function
Christhian Fonseca,Tiago Tavares 125

Comparing Meta-Classifiers for Automatic Music Genre Classification
Vítor Shinohara,Juliano Foleiss,Tiago Tavares 131

A chord distance metric based on the Tonal Pitch Space and a key-finding method for chord annotation sequences
Lucas Marques 136

Predicting Music Popularity on Streaming Platforms
Carlos Soares Araujo,Marco Cristo,Rafael Giusti . . . 141

Studio and Laboratory Reports

NESCoM Research Report (2019)
Leandro Costalonga,Daniel Coura,Marcus Vinícius Neves,Fabiano Costa,Helder Rocha . . . 151
Computer Music research at FEEC/Unicamp: a snapshot of 2019
Tiago Tavares,Bruno Masiero . . . 155
Alice: Arts Lab in Interfaces, Computers, and Everything Else - Research report (2019)
Flávio Schiavoni,André Gomes,João Teixeira,Frederico Resende,Igino Silva Junior,Gabriel Lopes Rocha,Avner Paulo,Igor Andrade,Mauro César Fachina Canito,Rômulo Augusto Vieira Costa . . . 157
A retrospective of the research on musical expression conducted at CEGeME
Maurício Loureiro,Tairone Magalhaes,Davi Mota,Aluizio Oliveira,Thiago Campolina . . . 165
MusTIC: Research and Innovation Group on Music, Technology, Interactivity and Creativity
Filipe Calegario,Giordano Cabral,Geber Ramalho . . . 173
LCM-Ufrgs Research Group Report: What are we doing in Computer Music?
Marcelo Pimenta,Marcelo Johann,Rodrigo Schramm . . . 181
Computer Music Research Group - IME/USP Report for SBCM 2019
Paulo Vitor Itaboraí de Barros . . . 189

Posters

A cluster analysis of benchmark acoustic features on Brazilian music
Leonardo Ferreira,Estela Ribeiro,Carlos Thomaz . . . 194
The taste of scales and chords
Alan Taitz,Diego Shalom,Marcos Trevisan,Bruno Mesz . . . 197
Automatic onset detection using convolutional neural networks
Willy Cornelissen,Maurício Loureiro . . . 199
Harmonia: a MuseScore's plugin to teach music
João Teixeira,Flávio Schiavoni,Rogerio Constante,Octávio Deluchi . . . 201
Audio Encryption Scheme based on Pseudo-orbit of Chaotic Map
Eduardo Magalhães,Thiago Santos,Erivelton Geraldo Nepomuceno . . . 204
Digital Design of Audio Signal Processing Using Time Delay
Mauricio Perez,Rodolfo Coelho de Souza,Regis Rossi Alves Faria . . . 207
3A: mAchine learning Algorithm Applied to emotions in melodies
Claudio Gomes,Josue da Silva,Marco Leal,Thiago Nascimento . . . 210
Instrumental Sensibility of Vocal Detector Based on Spectral Features
Shayenne Moura,Marcelo Queiroz . . . 212
Characterization of the sonority associated to woodwinds instruments through spectral analysis
Yubiry González,Ronaldo Prati . . . 215
Batebit Controller: Popularizing Digital Musical Instruments' Technical Development Process
Filipe Calegario,João Tragtenberg,Giordano Cabral,Geber Ramalho . . . 217

Art

Tecnofagia: A Multimodal Rite
Luzilei Aliel,Rafael Fajiolli,Ricardo Thomasi . . . 221
Graph Composer: music composition from graph design
Pedro Arthur,Vitor Rolla,José Eduardo Ayres,José Ezequiel Sánchez . . . 223
Per(sino)ficação
Fabio Carvalho,Flávio Schiavoni,João Teixeira . . . 225
O Chaos das 5
Flávio Schiavoni,Igino Silva Junior,João Teixeira,André Gomes,Frederico Resende,Gabriel Lopes Rocha,Avner Paulo,Igor Andrade . . . 227
Body Building Music: The Kinase Instalation
João Monnazzi,Regis Faria . . . 229
Black Lives Matter
Avner Paulo,Carlos Eduardo Oliveira de Souza,Bruna Guimarães Lima e Silva,Flávio Schiavoni,Adilson Siqueira . . . 231
Iterative Meditations: The use of audio feature extraction tools on acousmatic composition
Aluizio Oliveira . . . 233

Workshops

Introduction to automatic audio classification
Tiago Tavares 237
Procedural Music in Games
José Eduardo Ayres,Pedro Arthur,Vitor Rolla,Luiz Velho 239

Keytalks

The First Brazilian Symposium on Computer Music presents Brazilian computer music potentials - Caxambu, MG, 1994
Maurício Loureiro 243
The Politics of Computer Music
Fernando Iazzetta 248

Index

Authors 254
Institutions 255
Keywords 256

Full Papers

A computer-based framework to analyze continuous and discontinuous textural works using psychoacoustic audio descriptors

Micael Antunes [1, 2] **, Danilo Rossetti** [1, 2] **, Jônatas Manzolli** [1, 2]

[1] Arts Institute – Campinas University
Rua Elis Regina, 50 – Cidade Universitária "Zeferino Vaz", Barão Geraldo, Campinas - SP

[2] Interdisciplinary Nucleus for Sound Communication- Campinas University
Rua da Reitoria, 165 – Cidade Universitária "Zeferino Vaz", Barão Geraldo, Campinas, SP

micael.silva@nics.unicamp.br, danilo.rossetti@nics.unicamp.br, jonatas@nics.unicamp.br

Abstract

This paper discusses a computer-aided musical analysis methodology anchored on psychoacoustics audio descriptors. The musicological aim is to analyze compositions centered on timbre manipulations that explore sound masses and granular synthesis as their builders. Our approach utilizes two psychoacoustics models: 1) Critical Bandwidths and 2) Loudness, and two spectral features extractors: 1) Centroid and 2) Spectral Spread. A review of the literature, contextualizing the state-of-art of audio descriptors, is followed by a definition of the musicological context guiding our analysis and discussions. Further, we present results on a comparative analysis of two acousmatic pieces: *Schall* (1995) of Horacio Vaggione and *Asperezas* (2018) of Micael Antunes. As electroacoustic works, there are no scores, therefore, segmentation and the subsequent musical analysis is an important issue to be solved. Consequently, the article ends discussing the methodological implication of the computational musicology addressed here.

1. Introduction

Research in the late 20th century intensified the development of computer aid analysis using computer-integrated digital signal processes. There are studies in machine listening [1], auditory scene analysis [2], and music information retrieval [3]. In line with such trajectory, we discuss here an application of so-called Audio Descriptors (AD) to a musical analysis of textural works. Audio Descriptors (AD) comprises a series of algorithms, that uses statistical and mathematical models to analyze audio data using acoustics, psychoacoustics and/or musical parameters. It allows, for example, to perform a classification of pieces and musical styles [4] and musical information retrieval and classification such as in Peeters [4], Pereira [5], and Peeters et al. [6].
Additionally, Malt & Jourdan [7] and [8] have developed tools for the analysis of contemporary music. Rossetti & Manzolli [9] studied emergent sounds in the context of electroacoustic and live electronic music. Other approaches have been developed in the Interdisciplinary Nucleus for Sound Studies (NICS-UNICAMP). Monteiro [10] developed a methodology to automatic segmentation of electroacoustic music and Simurra & Manzolli [11] and [12] applied audio descriptors in the context of computer-assisted orchestration. In line with these previous studies, our main question here is: How can psychoacoustic descriptors contribute to an analytical methodology in the context of acousmatic compositions focused on textures?

To solve this question, we introduce and apply two audio descriptors: Volume [13] e Bark Coefficients [14]. These tools provide multidimensional graphical representations centered on two aspects: a) Psychoacoustics models like Loudness, and Critical Bandwidths and b) Spectral measures like Spectral Centroid, Spectral Spread and Loudness.

Based on these descriptors, we discuss the proposed methodology of analysis and compare two acousmatic works: *Schall* (1995) of Horacio Vaggione and *Asperezas* (2018) of Micael Antunes. Previous discussions addressing the analysis of those works are found in [15], [16], [17], and [18], here we mainly discuss the methodology and the computational tools employed in the process.

2. Review of literature

2.1 Musical Analysis and Psychoacoustics

Musical analysis based on psychoacoustic models had been particularly studied in recent years. Parncutt [19] analyzed tonal music using psychoacoustic features, based on pitch perception, fusion, and harmonicity models. Sethares [20] investigated tuning systems based on sensorial dissonance models applied in the analysis of Thai Classical Music. Sethares also developed a computer tool to generate dissonance curves aiming to calculate dissonance of sound spectra. Vassilakis [21] implemented an algorithm to calculate the roughness of audio files.

Psychoacoustic measures are useful tools regarding the study of pitch, timbre and harmony perception. In this context, Monteiro [10] implemented the Critical Bandwidth model in an external library called

PDescriptors[1] for Pure Data. Bullock [14] implemented Zwickers' model [22] in a Sonic Visualiser software plugin to generate plots on the levels of Critical Bandwidth energy of audio files (see analytical plots in topic 4).

Other psychoacoustics models are also implemented in libraries of Sonic Visualiser, Pure Data and Max/MSP, such as Sharpness, Loudness, Pitch Detection, and Inharmonicity[2].

2.2 Sound Mass Music

The texture is a metaphor of visual and tactile perception. This concept offers a particular standpoint aiming to describe and analyze some musical phenomena [23], [24] and [25]. According to Mackay [23], "texture of music consists of its sounding components; it is conditioned in part by the number of those components sounding in simultaneity or concurrence". Ferraz [24] emphasizes that texture is "compatible with the system and typical procedures to which it belongs". Texture can be polyphonic, monophonic, harmonic, serial, pointillistic or static. He argues that textural perception is correlated with the Gestalt phenomenon [26] resulting from the interaction between the compositional sound material.

We can analyze musical texture starting with the measure of musical events that are found in the time-frequency space, giving us a measure of density [25]. We are particularly interested in sound mass texture. This phenomenon occurs when the music texture is the protagonist of the musical discourse. Further, other musical parameters (pitch, rhythm, intensity) are just parameters to create a "kind of spectromorphological hyper-instrument" [27].

Karlheinz Stockhausen stated that the perception of sound mass starts at our limit of perceiving separated musical sounds in time or frequency space [28]. This idea was explored by Ligeti with the concept of "timbre of movement" (*bewegungsfarbe*) [29]. Stockhausen's concept of sound mass could be related to the technique of micropolyphony, in which the perception of a sound texture is created by the fusion of multiple separated instrumental voices [29]. Pieces like *Atmosphères* (1961), *Continuum* (1968) or *Chamber Concerto* (1970) are examples of this concept. In the context of Ligeti's electronic music (*Glissandi*, 1957, *Artikulation*, 1958) the instrumental voices are replaced by multiple sound partials, that, juxtaposed and superimposed by similar processes to additive synthesis, form the timbre of movement.

Another theory, *Stochastic Music,* was developed by Iannis Xenakis in the 1950s [30]. Using mathematical models, Xenakis aimed to create continuous and discontinuous sound masses using a large number of *pizzicati, glissandi* or even electronic sounds [30]. He explored this technique in works such as *Pithoprakta* (1955-56), *Achorripsis* (1958) and *Aroura* (1971).

The granular synthesis technique is also an important technique that generates sound masses. This process aims to generate sounds by the agglutination of thousands of 'grains', which are extremely short sound samples, belonging to the micro-time scale (in milliseconds) [31]. Perceptually, it is need more than 20 grains per second to build a structure that is perceived as a sound event. When this fusion happens, a texture or a flux of a timbre varying in time is generated. Solomos [32] assumes that the Granular Paradigm corresponds to the corpuscular sound description, as opposed to the waveform description.

From the definitions presented above, it is possible to assume that sound masses create a kind of saturation in the auditory system. At a certain point in time, the listener starts to perceive sound events globally, i.e. a unique timbre rather than separated events [33]. Therefore, we claim here is that psychoacoustic descriptors offer a proper tool to analyze such phenomenon. Critical Bandwidths and Loudness fearture extraction can be a useful tool to describe: a) how sound masses are formed and b) how they merge into a single perceived unity. Therefore, our analysis focus on these two aspects.

3. A Method of Computer Aid Analysis

Our method concerns the generation of plots of audio descriptors to describe spectral and psychoacoustic features from audio files (i.e. recordings of performance or acousmatic pieces). Those graphics are used to reveal important features of textural music. They are complemented by musicological discussions and further music analysis.

3.1 Volume Descriptors

One of our tools is a multidimensional descriptor comprising three curves in one graphic called Volume (Fig. 1 and 2, above). The volume is a concept defined by Truax [33] as the "perceived magnitude of sound or the psychological space it occupies".

Here, the Volume is a three-dimensional graphic representation created by Malt and Jourdan [13] by superposing three curves: Spectral Centroid, Spectral Spread, and Loudness. In the graphics, the Volume magnitude is given by the area between Spectral Centroid and Spectral Spread curves. Degrees of grey tonality indicates the Loudness of the perceived sound intensity. The dark is the higher level, while soft gray the is the lower.

The spectral centroid *C(i)* can be defined as the barycenter of the spectral energy concerning to a window of analysis [5], [10]. It is calculated as the frequency weighted mean, where *Xi(k)* are the magnitudes of the Discrete Fourier Transform of the *i* window, and *k* is the half of the adopted spectral components of the Transform. The spectral centroid *C(i)* is calculated as follows:

[1] https://sites.google.com/site/pdescriptors/ Accessed in 03 May 2019.

[2] https://www.vamp-plugins.org/download.html Accessed in 03 May 2019.

$$C_{(i)} = \frac{\sum_{k=1}^{K} k*|X_i(k)|^2}{\sum_{k=1}^{k}|X_i(k)|^2} \quad (1)$$

where *Xi(k)* for *k = 1...K* is the frequency amplitude of the window of analysis.

According to Peeters [4], the Spectral Spread *S(i)* describes the spread of the sound spectrum around its mean value, i.e. the variance of the defined distribution. Therefore, it is a measure of the average spread of the spectrum concerning its centroid. For noisy sounds, the Spectral Spread is expected to be high, while harmonic and tonal sounds, it is low. The Spectra Spread is defined as follows:

$$S_{(i)} = \frac{\sum_{k=1}^{K}(k-C_i)^2|X_i(k)|^2}{\sum_{k=1}^{k}|X_i(k)|^2} \quad (2)$$

where *Xi(k)* for k = 1...K is the frequency amplitude of the window of analysis.

Pereira [5] and Monteiro [10] define the Loudness *L(i)* as the psychoacoustic measure of the sound intensity perception, which varies in frequency according to the Fletcher and Munson curves [34]. Equation 3 defines the loudness *L(i)* of a spectral window of analysis, according to Pereira's model. The Fletcher and Munson curves are included in equation 3 as the W(k) weight function described in equation 4.

$$L_{(i)} = \sum_{k=1}^{K}|X_i[k]|^2 10^{w[k]/20} \quad (3)$$

where *Xi(k) for k = 1...K* is the frequency amplitude of the window of analysis.

$$W(k) = -0{,}6 * 3{,}64 * f(k)^{-0{,}8} - 6{,}5 * e^{-0{,}6*(f(k)-3{,}3)^2} + 10^{-3}*f(k)^{3{,}6} \quad (4)$$

where the frequency *f(k)* is measured in kHz, defined as *f(k) = k.d*, the difference between two consecutive spectral bins in kHz.

Finally, the *Spectral Volume V(i)* of the i-nth window of the DFT. is defined by the graphic:

$$V(i)=\{C(i), S(i), L(i)\} \quad (5)$$

where C(i) is plot as the curve profile, S(i) is plot as the distance within the profile and L(i) is the colored scale.

3.2 Bark Coefficients Descriptor

Bark is a unit of critical bandwidths of Zwicker's model [22]. The model consists of a subdivision of the range of audible frequencies based on the physiological properties of the basilar membrane.

Each *bark* is usually smaller than the pitch interval of one-third-octave and each slot is indexed with a number of the unit *bark,* i.e. the critical-band number zero is associated with 0 Hz. A critical-band-index smaller than one Bark indicates that two tones belong to the same critical band [22]. Consequently, masking, beating, and roughness can occur among these tones inside the same band.

It is important to say that psychoacoustic measures are based on empirical tests, therefore, a further mathematical formulation is required. An example is the so-called Fletcher & Munson's Curve of Iso-Loudness [34], equation (4). Moreover, we have already studied the Bark Scale anchored in Traunmüller [35] and others [36]. Our contribution is to present a new formula to convert frequency into barks.

We define a conversion function, B(*f*), divided into three curves and a scaling factor b(*f*). Nevertheless, our definition follows Traunmüller [35], clearness is improved.

Given a value $0hz \le f \le 20\ kHz$, the respective Bark Scale conversion function B(*f*) is calculated as follows:

$$B(f) = \begin{cases} b(f) + 0.15(2 - b(f))\,if\ b(f) < 2 \\ b(f) + 0.22(\,b(f) - 20.1)\ if\ b(f) > 20.1 \\ b(f)\ otherwise \end{cases} \quad (6)$$

where the scaling factor b(f) is given by

$$b(f) = \frac{26.81f}{(1960+f)} - 0.53 \quad (7)$$

In a previous study [18], we have used Bullock's [14] implementation to calculate the *bark coefficients* (e.g. Fig. 1 and 2, below). This descriptor divides the audible frequency range into 26 bark slots, using Zwicker's[3] model. It provides a measure of the sound energy into 26 slots, and it is quite similar to the Chroma audio descriptor[4].

Finally, the Bark coefficients and the Volume are used in three analytical tasks:

1) to create the piece segmentations using the Volume and Bark Coefficients;
2) to analyze harmonic fusion and segregation using Bark Coefficients;
3) examine the Sound Flux and the spectral evolution in time from the Volume representation.

4. Comparative Study

Here, we present a comparative study applying our analytical methodology in to three goals: 1) segmentaion, 2) fusion and segregation and 3) sound flux and loudness, as follows.

[3] Zwicker's [22] presents only 24 bark because the higher frequency of his table is 15500hz. But by using the equation to convert Hz in barks, we can calculate the 26 barks, containing all the range of audible frequencies.

[4] https://www.vamp-plugins.org/download.html Access 03 May 2019.

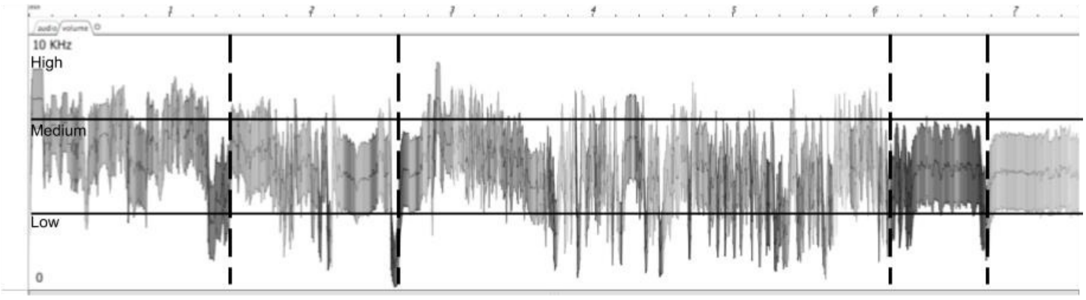


Figure 1: Plot of Volume *Schall* (1995). The vertical axis represents the frequency and the horizontal axis represents the time. The width of the line represents the spectral spread. The center line represents the spectral centroid. And the grayscale represents the loudness.

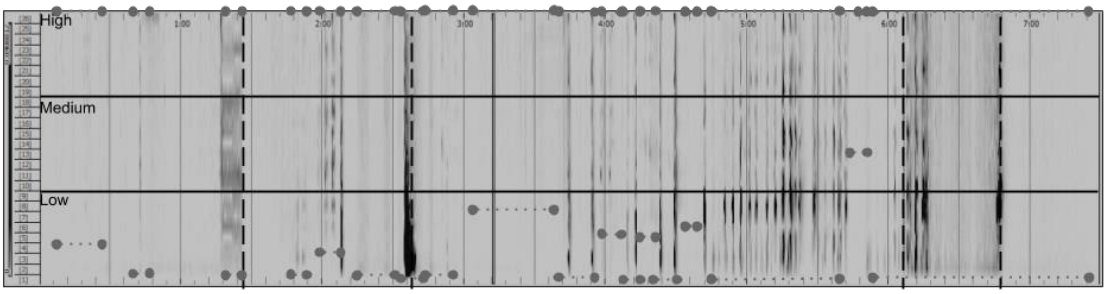


Figure 2: Plot of the Bark Coefficients of Schall (1995). The vertical axis represents the Bark Scale and the horizontal axis, time; grayscale, the energy of each bark bandwidth.

<u>Segmentation</u>: it is possible to segment a composition by observing peaks and valleys of energy, as well as breakpoints detected by in the Bark Scale and Volume descriptor.

<u>Fusion and Segregation</u>: it is possible to examine fusion or segregation levels of sound textures. These are related to the energy concentration of the Bark Scale.

<u>Sound Flux and Loudness</u>: it is possible to evaluated how much sound masses are not static applying the Spectral Spread and Loudness descriptors.

4.1 The Two Pieces

In this article, we analyze two works: *Schall* (1995) by Horacio Vaggione, and *Asperezas* (2018) by Micael Antunes. Since they are acousmatic works, there is no traditional musical score of the pieces. Thus, the provided segmentation an important feature addressed by the computer-aid analysis.

Schall (1995) is an acousmatic piece by Vaggione composed using the *micromontage*, an important compositional technique of the granular paradigm. The employed sound materials are piano recordings. These samples were edited in microtime domain, intending to create granular textures. *Schall* is a representative work concerning texture and sound mass manipulation of the granular paradigm.

Asperezas (2018), on the other hand, is a piece in which the compositional process is based on the threshold of beats and roughness sensations. These phenomena are related with the critical bandwidths energy manipulation. Thus, the piece was created using the FM synthesis technique, from a continuous timbre transformation in time [36]. *Asperezas* (2018) is anchored in a study of beats and roughness that is also correlated to our analytical model.

We compare both pieces under two main perspectives:

a) To employ the segmentation provided by the computer-aid analysis in order to describe the macroform of the works;
b) To enhance our musicological standpoint, comparing their analysis with the computer.

4.2 Comparing Descriptors segmentation

The pieces' segmentation was performed by employing together the Volume and Bark Coefficients. Peaks and valleys were taken into account in order to observe possible correspondences between them. In this sense, the information coming from both descriptors can be viewed as complementary, providing consistency in the segmentation of the piece.

Figures 1 and 2 show the Volume and Bark Coefficients of *Schall* (1995). The segmentation is represented by the dashed vertical lines. By this process, the piece was divided into 5 segments, see Table 1.

Section	1	2	3	4	5
Starting	0''	1'24''	2'39''	6'07''	6'49''

Table 1: Segmentation of *Schall* (1995), showing sections and its initial times.

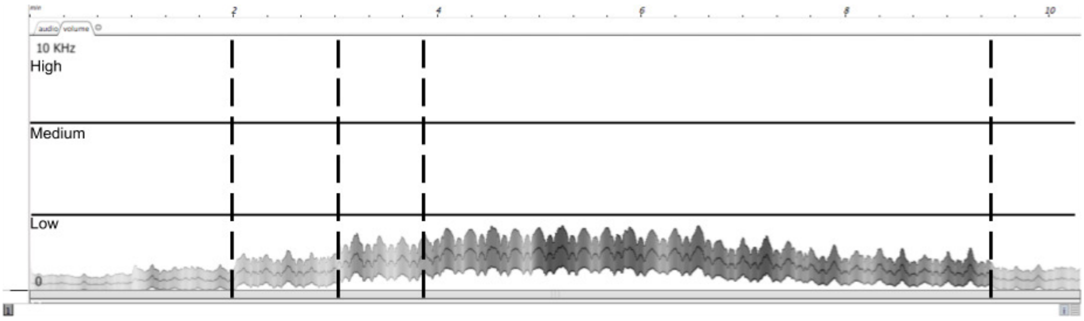


Figure 3: Volume of *Asperezas* (2018). The parameters are the same as in Figure 1.

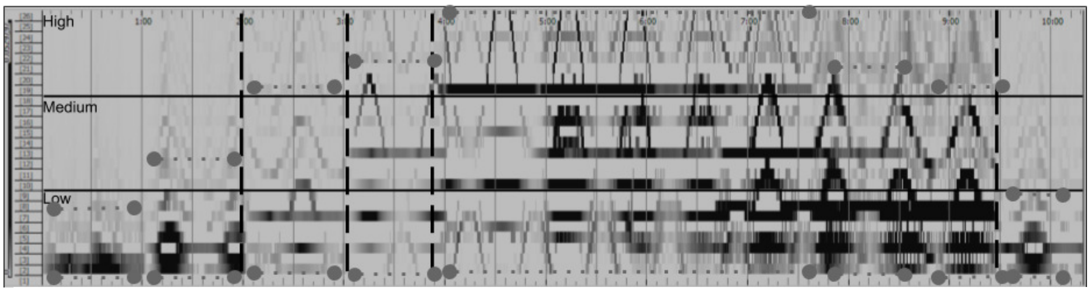


Figure 4: Plot of the Bark Coefficients of *Asperezas* (2018). The parameters are the same as in Figure 2.

Figures 3 and 4 show the Volume and Bark Coefficients of *Asperezas (2018)*. The segmentation is represented by dashed lines. By this process, the piece was divided into 5 segments, see Table 2.

Section	1	2	3	4	5
Starting	0’’	2’02’’	3’05’’	3’55’’	9’22’’

Table 2: Segmentation of *Asperezas* (2018), showing sections and its initial times.

4.3 Comparing Harmonic Fusion and Segregation

The Bark Scale graphic of *Schall* shows different density levels of grains, which mostly appear in the higher bark position. Grains with different changes in density do not allow significant harmonic fusion. For instance, in the second segment of the plot (Figure 2), the lowest level of fusion is found, while in segment 4 we have a peak of fusion.

Fusion and segregation in *Asperezas* are observed in Figure 4, within a pattern of recurrences. Therefore, there are frequencies which are present in the whole piece. We can highlight some aspects of the segment 4, in Figure 4: at the same time that some barks have constant energy (at some points, from 2 to 11), there is significant energy variation in all of the other bark slots. The consequence is a sensation of segregation because we perceive concomitant continuous frequencies and sound mass movement in their background.

Another aspect is the proximity of the barks concentrations that increases according to the evolution of the piece in time. This proximity generates several interferences. The consequence of this proximity is a high sensation of roughness and beatings.

4.4 Comparing Volume (Sound Flux and Loudness)

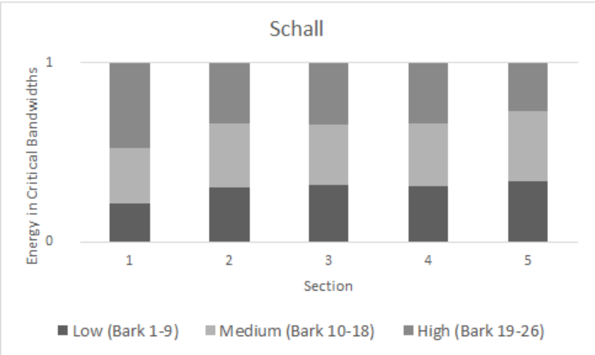
Schall (1995) belongs to the granular paradigm and it presents a considerable Sound Flux, i.e. a high level of grain’s movement in microtime, forming the sound texture or timbre of the work. The general level of loudness is low, except in the excerpt between 6’07’’ e 6’49’’, the fourth section of the piece, see Figure 1 and Table1.

We can observe in Figure 1 that the third section (which lasts around 3’30’’) presents a development focusing on the grain clouds’ manipulation and variations. The flux in this section has the higher level of the piece, reached by the considerable variations of grains’ intensity and frequency range. The first two sections present a considerable movement of sound clouds, but in a lower level compared to section 3. Sections 4 and 6 are more stable with lower sound flux, indicating an entropy movement inside the piece, such as the grains reaching a state of equilibrium. Therefore, in an overall view, the Spectral Volume in Schall (1995) is very diverse. The large Spectral Spread band indicates a preponderant timbre with noisy characteristics.

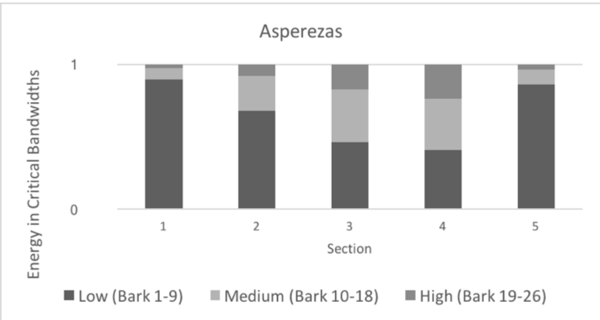
The sound flux of *Asperezas* (2018), on the other hand, has a clear direction. The development of the piece is linked with an expansion of the spectral frequencies and an increment of loudness (Section 4 has the peak of loudness). It is interesting to note that the spectral centroid of the volume representation increases at the same time that the bark concentration is higher.

4.5 *Schall* (1995) versus *Asperezas* (2018)

According to the intention of comparing the two pieces, we integrate the collected data of the bark coefficients descriptor in graphics 1 and 2. Thus, it is possible to have a global evaluation of the features extracted from the spectral development of the pieces. The graphics integrates the energy of the barks into three bands: Low (barks 1-9); Medium (barks 10-18); and High (barks 19-26). The energy of each band is ploted in Graphics 1 and 2 according to the sections of the pieces. The result is that *Schall* has its predominance in Medium and High bands while *Asperezas* has its predominance in the low bands.



Graphic 1: The energy of each section of *Schall* into bands of Bark Scale.



Graphic 2: The energy of each section of *Asperezas* into bands of Bark Scale.

We also integrated our analysis of Bark Coefficients, Spectral Fusion, Loudness and Spectral Flux in levels of variation: *Low, Medium and High*. Table 3 shows that *Schall* has a low variation on Bark; medium variation of Spectral Fusion and Loudness; and High variation of Spectral Flux. *Asperezas*, on the other hand, has a high variation in Bark and Spectral Fusion and low variation in Loudness and Spectral Flux.

	Bark	*Spectral Fusion*	*Loudness*	*Spectral Flux*
Schall	Low	Medium	Medium	High
Asperezas	High	High	Low	Low

Table 3: Level of variation of the parameters of analysis.

5. Discussions

The sound grains in *Schall* have some zones of concentration in each section. The montage of the grains in microtime domain reflects points of agglutination with denser textures and moments where the grains are scattered in the frequency space, generating a subtle texture with rarefied grain clouds.

In the micromontage technique applied in *Schall*, the idea is to achieve the saturation from the manipulation of sound particles, exploring our threshold of the pitch, timbre, and loudness perception. Thus, the synthesis, from the variation of levels of density, creates masses using sound particles, causing spectral and temporal masking phenomena. Thereby, starting with an amount of discontinuity, we create a global idea of continuity in the sound.

The macroform of *Asperezas* shows movement expressed by a spectral and loudness increasing, that decreases in the final of the piece. In the microform, the constant spectral oscillation is remarkable, such as the variations on the spectral fusion, and consequently changes of roughness and beatings.

In this sense, *Asperezas*, in the opposite of *Schall*, offers us a formal process that is higher predictable. The descriptors reveal a quasi-symmetrical transformation of the sound material (see figure 4) in the microform, creating in consequence a previous expectation in listening.

A comparative analysis of the two pieces raises an interesting compositional question. The sound masses are an effect of saturation of our auditory system. However, different techniques explore different thresholds of perception in different ways.

6. Conclusions

We discussed a methodology of computer-aided musical analysis using psychoacoustics audio descriptors that afford tools to study textural music. The aims of this study were to understand its singularities and enhanced hypothesis on the creative process. We compared two works *Asperezas* (2018) and *Schall* (1995) under three analytical perspectives: formal segmentation, harmonic fusion/segregation, and sound flux.

These questions are relevant in the musicological context concerning the analysis of contemporary music field, as they reveal the details of the compositional practices. They also allow us to study composition, not only from the standpoint of its creation, but to perform an analysis that gives us clues about the perceptual processes related to the listening of the works. Finally, audio descriptors are considered useful tool to support these issues.

In our analysis, we observed the singularities of the sound textures explored in the pieces. In this way, we emphasize the difference between the texture explored by the granulation paradigm, supported by the *micromontage* technique and the construction of the masses through the accumulation of sound phenomena in the critical bands. In the specific case of the two pieces, it was also observed the differences in the development of the pieces, by highlighting that the expectation of listening can act from the manipulation of the sound material.

Since the analysis is performed from the audio data, it would be possible to conceive other experiments using different kinds of recordings. For example, to evaluate different recordings of a piece using dummy head microphones in order to examine perceptual issues. The approach introduced here can also be explored in the context of live electronic and instrumental music, aiming to analyze textural music by Ligeti, Xenakis, and Penderecki, for example.

We also foresee applications in computer-aid composition, the psychoacoustic descriptors would be useful in sound designing of interactive performances and multimodal installations. They could be applied together with theories of ecological perception and dynamic cognition, creating immersive computer-controlled environments.

At least, computational musicology dialogues with experimental psychology and cognitive sciences, therefore, some of the future developments could be useful to implement algorithms of automatic classification.

7. Acknowledgments

Micael Antunes is supported by FAPESP in his doctoral research at Arts Institut of Unicamp/ NICS-UNICAMP, process 2019/09734-3. Danilo Rossetti is supported by FAPESP in his post-doctoral research at NICS-UNICAMP, process 2016/23433-8. Jônatas Manzolli is supported by CNPq under a Pq fellowship, process 304431/2018-4 and 429620/2018-7.

8. References

[1] Rowe, R. Interactive music systems: machine listening and composing. MIT Press, 1992.

[2] Bregman, A. S. Auditory scene analysis: The perceptual organization of sound. MIT Press, 1994.

[3] Herrera, P. MIRages: an account of music audio extractors, semantic description and context-awareness, in the three ages of MIR. PhD Thesis, Information and Communication Technologies, Universitat Pompeu Fabra. Barcelona, 2018.

[4] Peeters, G. A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project (Cuidado Project Report). Paris: Ircam, 2004.

[5] Pereira, E. Estudos sobre uma ferramenta de classificação musical. Dissertação de Mestrado. Faculdade de Energia Elétrica e de Computação da Universidade Estadual de Campinas, 2009. Campinas: UNICAMP, 2009.

[6] Peeters, G. et al. The Timbre Toolbox: Extracting audio descriptors from musical signals. Journal of the Acoustic Society of America, Melville, v. 130, n. 5, p. 2902-2916, November 2011.

[7] Malt, M; JOURDAN, Emmanuel. Z. Descriptors: A Library for Real-Time Descriptors Analysis. In: SOUND AND MUSIC COMPUTING CONFERENCE, 5, 2008, Berlin. Proceedings... Berlin: Sound and Music Computing, 2008.

[8] Malt, M; Jourdan, Emmanuel. Z. Real-Time Uses of Low Level Sound Descriptors as Event Detection Functions Using the Max/MSP Zsa.Descriptors Library. In: BRAZILIAN SYMPOSIUM ON COMPUTER MUSIC, 12., 2009, Recife. Proceedings... São Paulo: USP, SBC, 2009, p. 45-56.

[9] Rossetti, D.; Manzolli, J. De Montserrat às ressonâncias do piano: uma análise com descritores de áudio. Opus, 23(3), 193-221. 2017.

[10] Monteiro, A. Criação e performance musical no contexto dos instrumentos musicais digitais. Dissertação de Mestrado. Instituto de Artes da Universidade Estadual de Campinas, 2012. Campinas: UNICAMP, 2012.

[11] Simurra, Ivan; Manzolli, Jônatas. Sound Shizuku Composition: A Computer-Aided Composition System for Extended Music Techniques. MusMat Brazilian Journal of Music and Mathematics, v. 1, n. 1, p. 86-101, 2016.

[12] Simurra, Ivan; Manzolli, Jônatas. "O azeite, a lua e o rio": o segundo diário de bordo de uma composição a partir de descritores de áudio. Revista Música Hodie, Goias, v. 16, n. 1, p. 121-123, 2016.

[13] Malt, M; Jourdan, E. Le "BSTD"–Une représentation graphique de la brillance et de l'écart type spectral, comme possible représentation de l'évolution du timbre sonore. In: L`ANALYSE MUSICALE AUJOURD`HUI, CRISE OU (R)ÉVOLUTION, 2009, Strasbourg. Proceedings… Strasbourg: STAM, 2009.

[14] Bullock, J. Libxtract: a lightweight library for audio feature extraction; Proceedings of the 2007 ICMC; ICMA, Sweden, 2007.

[15] Rossetti, D. Temporalidades, morfologias e granulações em *Schall* de Horacio Vaggione. Proceedings of the *IV Encontro Internacional de Teoria e Análise Musical,* Eitam 2017, São Paulo, SP, 2017.

[16] Rossetti, D; Manzolli, J. Timbre emergente na música eletroacústica: Uma análise com o aporte gráfico do volume e da vivacidade espectral. Proceedings of the XXVIII Encontro da Sociedade Brasileira de Acústica. Porto Alegre, Unisinos, 2018.

[17] Rossetti, D; Manzolli, J. Analysis of Granular Acousmatic Music: Representation of Sound Flux and Emergence. *Organised Sound* v. 24, n. 2, p. 1-12, 2019.

[18] Silva, M. A., Botechia, T., Rossetti, D. Manzolli, J. Análise musical por descritores de áudio baseados em modelos psicoacústicos. Anais do Simpósio Internacional de Cognição e Artes Musicais 14. Campo Grande, MS, 2019. in press.

[19] Parncutt, R. Harmony: A psychoacoustical approach. Vol. 19. Springer Science & Business Media, 2012.

[20] Sethares, W. A. Tuning, timbre, spectrum, scale. Berlin: Springer Science & Business Media, 1998.

[21] Vassilakis, P. N.; Fitz, K. SRA: A web-based research tool for spectral and roughness analysis of sound signals. In: Proceedings of the 4th Sound and Music Computing (SMC) Conference, 2007. p. 319-325.

[22] Zwicker, E. Subdivision of the audible frequency range into critical bands (Frequenzgruppen). The Journal of the Acoustical Society of America, v.33, n.2, p.248-248,

1961.

[23] Mackay, J. On the perception of density and stratification in granular sonic textures: An exploratory study. Journal of New Music Research, v. 13, n. 4, p. 171-186, 1984.

[24] Ferraz, S. Análise e Percepção Textural: Peça VII, de 10 Peças para de Gyorgy Ligeti. Cadernos de Estudo: Análise Musical, vol. 3. São Paulo, pág. 68-79, 1990.

[25] Berry, W. Structural functions in music. Chelmsford: Courier Corporation, 1987.

[26] Cambouropoulos, E. Tsougras, C. Auditory Streams in Ligeti's Continuum: A Theoretical and Perceptual Approach. Journal of interdisciplinary music studies, v. 3, n. 1-2, p. 119-137, 2009.

[27] Smalley, D. Spectromorphology: explaining sound-shapes. Organised sound, Cambridge, v. 2, n. 2, p. 107-126, 1997.

[28] Stockhausen, K. Pontos e grupos. Im: Maconie, Robin. Stockhausen sobre a música: palestras e entrevistas compiladas por Robin Maconie. Tradução de Saulo Alencastre. São Paulo: Madras, 2009, pág. 45-50.

[29] Ligeti, G. Neuf essais sur la musique. Genebra: Éditions Contrechamps, 2010.

[30] Xenakis, I. Concerning time. Perspectives of New Music, Seattle, v. 27, n. 1, p. 84-92, 1989.

[31] Roads, C. Microsound. MIT Press. Massachusetts. 2001.

[32] Solomos, M. 2013. De la musique au son: l'émergence du son dans la musique des XXe-XXIe siècles. Rennes: Presses Universitaires de Rennes.

[33] Truax, B. Musical Creativity and Complexity at the Threshold of the 21st Century. Interface v. 21, p. 29-42, 1992.

[34] ISO, B. 226: 2003: Acoustics–Normal equal-loudness-level contours. International Organization for Standardization, v. 63, 2003.

[35] Traunmüller, H. "Analytical expressions for the tonotopic sensory scale", in The Journal of the Acoustical Society of America 88.1, 1990.

[36] SILVA, M. A. Redução da Dissonância Sensorial em uma Escala Temperada Utilizando Timbres Inarmônicos: Uma Abordagem Experimental e Aplicações Artísticas. Master Dissertation, 2018. 96 p. Music post-graduation program- School of Arts and Communications, São Paulo University, São Paulo, 2018.

J-Analyzer: A Software for Computer-Assisted Analysis of Antônio Carlos Jobim's Songs

Carlos Almada[1], **João Penchel**[1], **Igor Chagas**[1], **Max Kühn**[1], **Claudia Usai**[1], **Eduardo Cabral**[1], **Vinicius Braga**[1], **Ana Miccolis**[1]

[1]MusMat Research Group – School of Music – Federal University of Rio de Janeiro
Av. Chile, 330/21st floor – 20031-170 – Rio de Janeiro, RJ

carlosalmada@musica.ufrj.br

Abstract. *The present paper describes structure and functioning of J-Analyzer, a computational tool for assisted-analysis. It integrates a research project intended to investigate the complete song collection by Brazilian composer Antônio Carlos Jobim, focusing on the aspect of harmonic transformation. The program is used to determine the nature of transformational relations between any chordal pair of chords present in a song, as well as the structure of the chords themselves.*

1. Introduction

This paper is associated with a research project entitled "Systematic studies in popular music", coordinated by Carlos Almada, and developed with the cooperation of seven graduate and undergraduate students, members of the research group MusMat, in the ambit of Federal University of Rio de Janeiro. It describes the structure, functioning, and outputs of a computational program, named *J-Analyzer*, employed as auxiliary tool in the analysis of the complete song collection written by Brazilian composer Antônio Carlos Jobim.

The analysis of Jobim's songs is focused on two specific aspects: binary relations between contiguous chords and the structure of the chords themselves. Some basic questions motivated the elaboration of the project: (a) Which are the most recurrent and, especially, the most peculiar connections between chords used by Jobim? (b) These choices could be due to (at least, partially) voice-leading issues? (c) Which are the chordal qualities employed in this repertoire, considering also cardinality variations? (d) How is their distribution? A in-depth analytical, and original investigation considering these questions (among others) aims at expanding the knowledge in respect to the particular Jobim's harmonic conception, shedding lights also to a neglected field in systematic approaches, namely, structural studies in popular music.

2. Theoretical framework and background

The project is theoretically grounded in formulations derived from the Transformational and Neo-Riemannian theories (especially, [1], [2], and [3]), which based an original system (S-J), whose structure was recently described in an international conference on musical theory and analysis [4].[1] S-J intends to model Jobim's harmonic practice, through a formal approach addressing both chordal structures and binary relations. Basically, the system operates through the notion of *classes of chordal quality*, or more simply, chordal classes. Basically the classes are archetypical, abstract structures that represent tonal chords through correspondences with their internal structures, which define their respective qualities. We consider ten classes of chordal qualities: major with major seventh (labelled as Z),[2] dominant seventh (Y), French-sixth[3](X), augmented-fifth dominant (W), minor with seventh (z), half-diminished (y), diminished seventh (x), minor with major seventh (w), as well as major (V) and minor triads (v).[4] These basic classes in turn contain subsets formed by variants (for example, the dominant qualities "with flat nine", "with thirteenth", and "substitute fourth" are three members of the class Y).

The classes are derived through application of a group of special functions, algebraically, and geometrically[5] formalized. The subclasses are derived from the classes through application of transformational operations.[6]

3. Pre-analytical considerations

The analysis of each song is preceded by four stages:

- Transcription of the vocal and piano score[7] as a file in MusicXml format;
- Revision of the transcription, searching for errors and incongruities of any type, concerning the harmonic aspect;
- Rewriting of the chords (maintaining their original metric positions) as *normal-form structures*, i.e., in the most compact disposition (inside one octave), and always in root position;
- Storing of the normal-form harmonic sequence as a midi file, for further analysis in the program J-Analyzer.

The scheme of Figure 1 summarizes the four stages of the pre-analytical process.

[1]For some studies related to a previous version of the current system, see [5], [6], and [7].

[2]We adopt the final letters of the alphabet, in reverse order, for labelling the classes in order to not confuse with the initial letters (A-G), employed for naming chordal roots.

[3]Or dominant chord with flat fifth.

[4]Both classes are left at the end of the list because triads are relatively uncommon in Jobim's harmony.

[5]The geometric projections of the chordal classes is made on an original planar scheme, named *Tetranetz*.

[6]A detailed explanation about theoretical aspects of S-J is presented in [8].

[7]We adopt as reference for the transcription process the five volumes of the *Cancioneiro Jobim* [9].

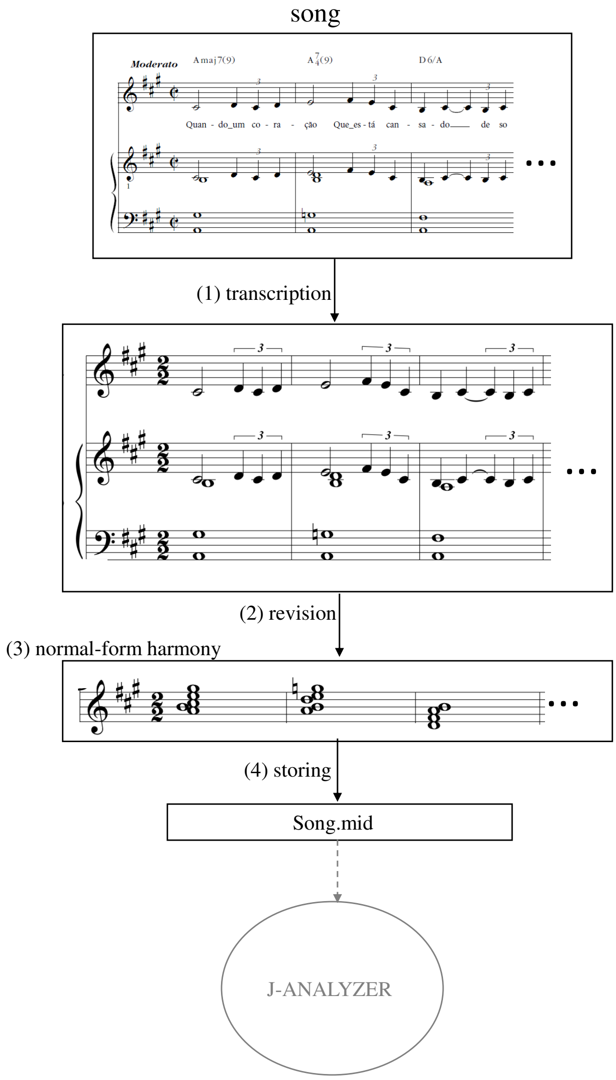


Figure 1: Representation of the four pre-analytical stages considering one of Jobim's song.

4. Functioning and structure of J-Analyzer

J-Analyzer was implemented in Matlab language, encompassing a set of complementary and/or sequential embedded functions and scripts. Since binary relations is one of the central aspects concerned in the analytical process, J-Analyzer works with a two-chord window per turn (i.e., chords 1-2, 2-3, 3-4, ..., (n-1)-n). For each pair of chords selected J-Analyzer returns their pitch-class content, chordal cardinalities (i.e., the number of elements), the set of common pitch classes, roots (expressed also as pitch classes), root distance (as number of semitones, between -5 and 6), class/subclass formal labels, and finally a binary-transformational operational designation, which expresses precisely, and concisely not only the relation between the two chords, but also their internal structure. Each chordal class or subclass registered in the system is named according to a simple convention: instances of the basic ten classes, above described, are labelled with the corresponding letter followed by a ”zero” (for example, Z0, representing any major chord with major seventh, does not mattering which specific root is in question); variants of the classes (i.e., subclasses) are identified by a sequential number different from zero, associated with the corresponding letter (e.g., Z1, for the subclass that represents all major chords with major seventh and major ninth). Figure 2 presents a basic overview of the analytical output, taking as example the chordal progression DMaj7–Gm.

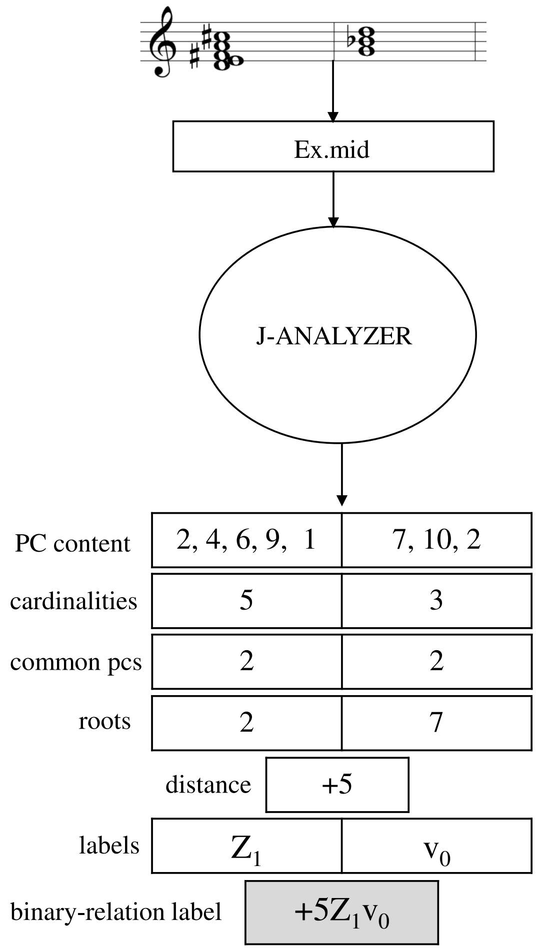


Figure 2: Example of output of J-Analyzer.

The functioning of J-Analyzer is essentially simple and can be described as a sequence of some steps, following the opening of the midi file corresponding to the song to be analyzed:

- Conversion of the midi information into a numeric matrix (or *nmat*) *i* x 7,[8] where *i* corresponds to the number of events present in the song (Figure 3 shows a short excerpt of the nmat corresponding to the normal-form harmony of *Garota de Ipanema* (Tom Jobim and Vinícius de Moraes), as illustration). The columns refer to, respec-

[8] A nmat is a numeric matrix produced by the function *midi2nmat* (which integrates a MIDI toolbox created by the researchers Tuomas Eerola and Petri Toiviainen, from Jyväskylä University, Finland).

tively, (1) beat positions of the onsets (considering 1 as a quarter note and 0 as the starting point), (2) durations (in beat multiples or sub-multiples), (3) track numbers, (4) pitches (in midi-pitch numbers), (5) velocity, (6) time points (measured in seconds), and (7) durations (also in seconds);

(1) onset (beats)	(2) duration (beats)	(3) track	(4) pitch	(5) velocity	(6) onset (seconds)	(7) duration (seconds)
0	4	3	65	69	0	2
0	4	3	69	69	0	2
0	4	3	72	69	0	2
0	4	3	76	69	0	2
8	4	3	67	70	4	2
8	4	3	71	70	4	2
8	4	3	74	70	4	2
8	4	3	76	70	4	2
8	4	3	77	70	4	2
12	4	3	60	70	6	2
12	4	3	61	70	6	2
12	4	3	65	70	6	2
12	4	3	67	70	6	2
12	4	3	70	70	6	2

• • •

Figure 3: First fifteen lines of *Garota de Ipanema* (Jobim and Moraes) harmonic nmat.

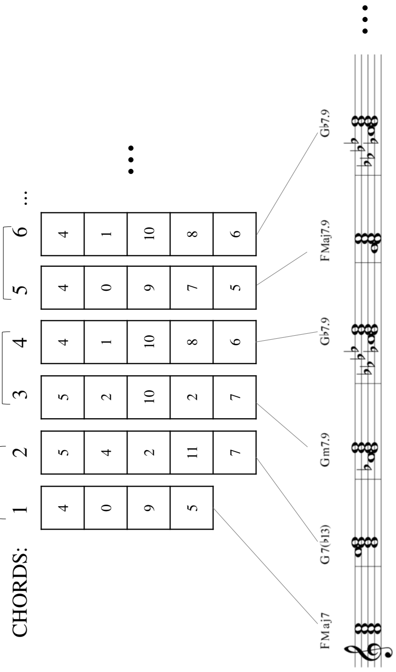


Figure 4: First six chords of *Garota de Ipanema* in pitch-class notation and normal form. Brackets indicate pairs of chords that will be considered for the subsequent analysis.

- Because only pitch and onset information are relevant in the analysis, both columns 1 and 4 are extracted from the nmat and merged in a new matrix of dimensions ix2. Pitches are then converted into pitch classes (through simple modulo 12 operation). Since the exact position of the chords does not matter for the analysis, but their relative positions (first, second, etc.), a new matrix is created, depicting vertically, and sequentially the chords as pitch classes in normal-form disposition (Figure 4);
- The program selects then the two first chords for analysis. Roots, cardinalities and pitch-class contents are automatically obtained. After stored (for further statistical purposes), each normal-form ordered set is converted into a *prime form*, by subtracting in mod12 the pitch-class root from its content (or else, for convention, any prime form has root transposed to pitch class 0). The intention behind this normal-prime convertion is to obtain a still more abstract representation of each chord, which fits the goals of the analytical investigation. The intervallic configuration of the prime form is then codified as an unique integer by the function ”pcset2Goedel”, which implements an adaptation of the algorithm known as *Gödel Numbering*. Initially, this algorithm arbitrates for each symbol of a string formed by *n* typografic signs (originally, a logical proposition of the Theory of Numbers) a numeric code. The codified, ordered numbers (that substitute for the string's symbols) are then considered as exponents raising a sequence of the *n* first prime numbers. The product of the raised primes results in a (normally very big) integer, an univocal index[9] that, ultimately, represents the original proposition (its factoring returns exactly the same sequence of code numbers, which can then be translated to the initial symbols).[10] In our system S-J, for a Jobinian prime-form chord of cardinality *n*, instead of using directly their n pitch classes, the algorithm extracts the sequence of their intervals, forming a vector of n-1 elements, treated as the exponents that raise the sequence of n-1 first primes (because the intervals are expressed directly as numbers, the phase of translation of symbols is here bypassed). There are two good reasons for use intervals, rather than pitch classes as exponents: (1) conciseness, due to the use of one less element, and especially lower-value ex-

[9]This argument is based on the *Fundamental Theorem of Arithmetic*, which states that any integer greater than 1 is a prime number or it is the result of the product of prime numbers.

[10]This extremely elegant and clever algorithm was elaborated by Austrian mathematician and logician Kurt Gödel (1906-1978), associated with which is now known as his *Incompleteness Theorems*. For more detailed information about Gödel, his algorithm, and the revolutionary aspect of his contributions to the mathematic logic, see [10]. For another musical application of Gödel Numbering, see [11].

ponents (both contributing for Gödel numbers of lower magnitude), and (2) the fact that the intervallic vector represents the very unique structure of the chord which with is associated, a merit in itself. The integer that results from the raised-prime product is considered the Gödel number (G) associated with the prime-form chord in question (a strictly one-to-one relation). In other words, there are in the system as many *unique* Gs as there are classes and subclasses of qualities. The two Gs are then compared with a previously-prepared lexicon of chordal class and subclasses, and their definitive formal labels are obtained. It is noteworthy mention that another incorporated function (pc2label) translates normal forms into chordal labels in informal notation (in the case of the example, "FMaj7.9" and "G7.13"), which are then displayed in an output list and stored for further statistical analysis. Figure 5 summarizes the sequence of events in this stage;

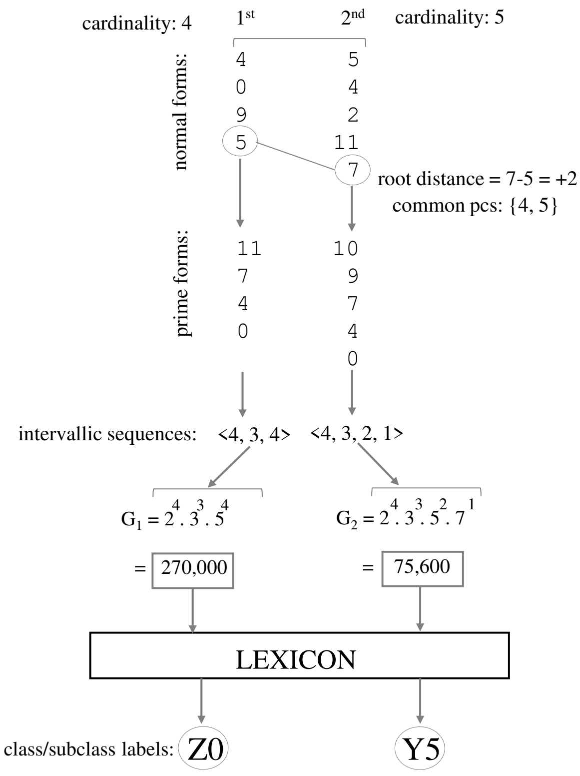


Figure 5: Outputs from the analysis of the first chordal binary relation of *Garota de Ipanema*: normal and prime forms, common pitch classes, roots, root distance, intervallic sequences, Gödel numbers, and class/subclass formal labels.

- Finally, the operation label related to the transformational connection of the two chords (or, in formal terms, the binary-transformational operation) is produced, being formatted as a string, by assembling root distance, and the two quality formal labels (preserving the same order of apparition of the chords). In the case exemplified, the operation is identified as the string +2Z0Y5. Similarly what was made for the intervallic sequences, the operation label is also converted in a Gödel number, which is then stored. However, this case has two particularities that deserve to be mentioned: (a) since the string has always the same cardinality of six fixed positions (direction-interval-letter-number-letter-number), the sequence of prime bases is consequently also fixed: $<$2, 3, 5, 7, 11, 13$>$; and (b) beside numbers, other symbols are also used in the string, which implies that a previous translation stage is necessary. Concerning this, codes were assigned to the non-numeric symbols used, as shown in Table 1. These symbols (that occupy the odd positions of the string) were subdivided according to the categories which with are associated: intervallic direction (0 or 1, respectively, for ascending or descending), seventh chords with major quality (Z to W, 0 to 3),[11] seventh chords with minor quality (z to w, 4 to 7), and triads (V and v, 8 and 9). Even positions are filled by numbers (i.e., interval sizes and order numbers of classes/subclasses), and therefore represent themselves in the translated sequence. Thus, the string corresponding to the first couple of chords of the example is numerically encoded as follows:
+2Z0Y5 $\equiv$ $<$0,2,0,0,1,5$>$, which become exponents for the six prime bases, from whose product is obtained the corresponding Gödel number $G_{1\text{-}2}$ (i.e., referred to the binary relation between chords 1 and 2):

$$G_{1\text{-}2} = 2^0.3^2.5^0.7^0.11^1.13^5 = 36,758,007$$

In spite of its magnitude, $G_{1\text{-}2}$ is stored by J-Analyzer (together with the remaining indices that represent the following binary relations). When necessary the information related to these huge numbers are easily retrieved through prime factoring and subsequent translation.

- After concluding the analysis of the first binary relation, another pair of chords (numbers 2 and 3, in this case) is then selected and the whole analytical process is repeated by the program, and so on, until the end of the harmonic progression (chords n-1 and n). The complete set of data is stored in matrices. A list summarizing the main information for the examination by the user/analyst is then produced as a visual output of J-Analysis (Figure 6 adapts a brief portion of the list referred to the analysis of *Garota de Ipanema*).[12]

[11] Because symbols for direction are exclusively settled at the first position of the string, codes 0 and 1 can be also attributed to the quality symbols, that form the third and fifth positions.

[12] Because of their great magnitudes, the Gödel numbers of prime forms and operations are not displayed in the list.

Table 1: Encoding of symbols used in the binary-transformational operations.

symbol	meaning	code
+	ascending interval	0
-	descending interval	1
Z	major w/ major seventh	0
Y	dominant seventh	1
X	French-sixth	2
W	augmented dominant	3
z	minor w/ seventh	4
y	half-diminished	5
x	diminished seventh	6
w	minor w/ major seventh	7
V	major triad	8
v	minor triad	9

5. Concluding remarks

This paper presented basic structure and functioning of the program J-Analyzer, a computer-assisted tool used in an in-depth, original analysis of Jobim's songs, addressing transformational harmonic aspects. With 110 songs already analyzed (out a total of 146), the first phase of the research is next to its conclusion. J-Analyzer can be considered as a powerful tool for the tasks to which was designed, namely, conversion of midi file into individual chordal information (essentially, structure, cardinality, and class/subclass type), and the transformational relations that connect each pair of them. More specifically, the strategy of adaptation Gödel-numbering function to the encoding of essential information for the analysis proved highly effective, since the compaction of numeric sequences and symbolic strings into univocal integers contributes to economy in the storing of data as well as facilitates retrieval processes.

As next stages of the present research, the multitude of data produced in the analyzes will be combined, crossed and interpreted, through the help of another software, J-Statistics, considering not only the complete repertoire of songs, but also their subdivision into the so-considered five creative periods of Jobim (1947-58; 59-65; 66-70; 71-82; and 83-94), in order to evidence tendencies, particularities, and intersections of his harmonic preferences.

chord	informal label	normal form	common pcs	root	root dist.	formal label	operation
1	FMaj7	5,9,0,4	–	5	–	Z0	–
2	G7.13	7,e,2,4,5	4,5	7	+2	Y5	+2Z0Y5
3	Gm7.9	7,9,t,2,5	2,5,7	7	+0	y2	+0Y5y2
4	Gb7.9	6,8,t,1,4	t	6	-1	Y1	-1y2Y1
5	FMaj7	5,9,0,4	4	5	-1	Z0	-1Y1Z0
6	Gb7.9(#11)	6,8,t,0,1,4	0,4	6	+1	Y8	+1Z0Y8
7	FMaj7.9	5,7,9,0,4	0,4	5	-1	Z2	-1Y8Z2
8	G7.13	7,e,2,4,5	4,5,7	7	+2	Y5	+2Z0Y5
9	Gm7.9	7,9,t,2,5	2,5,7	7	+0	y2	+0Y5y2
10	Gb7.9	6,8,t,1,4	t	6	-1	Y1	-1y2Y1
...	...	...	...	...	...	...	...

Figure 6: Partial view of the data-list of *Garota de Ipanema*, considering numbers of the chords, informal labels of the chords, normal forms, sets of common pitch classes, roots, root distances, quality formal labels, and binary-operation labels.

References

[1] LEWIN, D. (1987). **Generalized Musical Intervals and Transformations**. New Haven: Yale University Press.

[2] KOPP, D. (2002). **Chromatic Transformations in Nineteenth-Century Music**. New York: Cambridge University Press.

[3] RINGS, S. (2011). **Tonality and Transformation**. Oxford: Oxford University Press.

[4] ALMADA, C. etal (2019). An Analytical Methodology for Harmonic Analysis of Antônio Carlos Jobim's Songs. **International Symposium on the Analysis and Theory of Music**, 1., Proceedings. Porto.

[5] ALMADA, C. (2017a). Uma proposta teórica visando à aplicação de princípios neorriemanianos em música popular. **Congresso da Associação Nacional de Teoria e Análise Musical**, 4., Proceedings. Florianópolis: UDESC, 20–30.

[6] ALMADA, C. (2017b). Representação geométrica de conduções parcimoniosas de vozes emprogressões harmônicas em música popular. **Encontro Nacional da ANPPOM**, 27., Proceedings. Campinas: UNICAMP.

[7] KUHN, M. etal (2017). Relações neorriemanianas de acordes de sétima na primeira fase composicional de Antônio Carlos Jobim. **Encontro Internacional de Teoria e Análise Musical**, 4., Proceedings. São Paulo: USP, 147–183.

[8] ALMADA, C. etal (2018). Uma nova abordagem referente à ocorrência de transformações harmônicas em canções de Antônio Carlos Jobim. **Congresso Brasileiro de Música e Matemática**, 3., Proceedings. Rio de Janeiro: UFRJ (in press).

[9] JOBIM, A. (2006). **Cancioneiro Jobim**: Obras escolhidas. Rio de Janeiro: Instituto Antônio Carlos Jobim (Score).

[10] NAGEL, E. and NEWMAN, J. (2001). **Gödel's Proof.** New York: New York University Press.

[11] ALMADA, C. (2017b). Gödel-Vector and Gödel-Address as Tools for Genealogical Determination of Genetically-Produced Musical Variants. In: **The Musical-Mathematical Mind: Patterns and Transformations.** Cham (Switzerland): Springer , 9–16.

Identifying Narrative Contexts in Brazilian Popular Music Lyrics Using Sparse Topic Models: A Comparison Between Human-Based and Machine-Based Classification.

André Dalmora[1], **Tiago Tavares**[1 2]

[1]School of Electric and Computer Engineering (FEEC) – University of Campinas (Unicamp)
CEP 13083-852 – Av. Albert Einstein, 400 – Campinas – SP – Brazil

[2]Interdisciplinary Nucleus for Sound Studies (NICS) – University of Campinas (Unicamp)
CEP 13083-872 – Rua da Reitoria, 165 – Campinas – SP – Brazil

andre.dalmora@gmail.com

Abstract. *Music lyrics can convey a great part of the meaning in popular songs. Such meaning is important for humans to understand songs as related to typical narratives, such as romantic interests or life stories. This understanding is part of affective aspects that can be used to choose songs to play in particular situations. This paper analyzes the effectiveness of using text mining tools to classify lyrics according to their narrative contexts. For such, we used a vote-based dataset and several machine learning algorithms. Also, we compared the classification results to that of a typical human. Last, we compare the problems of identifying narrative contexts and of identifying lyric valence. Our results indicate that narrative contexts can be identified more consistently than valence. Also, we show that human-based classification typically do not reach a high accuracy, which suggests an upper bound for automatic classification.*

1 Introduction

Songs and their lyrics are cultural elements that are frequently linked to the perception of subjective experiences [1, 2]. They can be embedded into cultural activities such as playing games, working, dancing, storytelling, and fighting [3]. The emotional perception of songs is related to inherited cultural aspects of the listener [4]. Regardless of the musical context, the perception of emotions is linked to a person's personal history [5] and to their cultural background [6].

In contemporary Western popular songs, lyrics often refer to similar subjects, such as "a lost love" or "reflections about life". These subjects can be loosely related to the concept of *archetype*, [7, 8]. Archetypes are recurring behavior patterns that can be observed in several elements of the same domain, such as "hero", "sage", or "damsel in distress" in the literature.

The recurring themes of popular songs can also be interpreted as affect-related [9] classification tags. Under this interpretations, affects are described according to a situation presented to a protagonist (which is often the musician themselves) and their reactions related to that, similarly to Russell and Barrett's idea of *prototypical emotional episodes* [10].

In this work, we use text mining tools to automatically classify music lyrics into categories related to their narrative contexts. For such, we built a dataset containing Brazilian popular music lyrics which were raters voted online according to its context and valence. We approached the problem using a machine learning pipeline in which lyrics are projected into a vector space and then classified using general-purpose algorithms. We experimented with document representations based on sparse topic models [11, 12, 13, 14], which aims to find groups of words that typically appear together in the dataset. Also, we extracted part-of-speech tags for each lyric and used their histogram as features in the classification process.

Additionally, we evaluated the classification accuracy difference related to using valence-related categories [14] instead of narrative-based categories. Valence-related categories are more common than story prototypes in text mining [14, 15, 16, 17], and are more connected to subjective perceptions of the listeners.

In order to compare machine and humans we quantified how much humans disagreed with each other. This was made considering each human rater as a predictor of the most-voted labels and calculating their accuracy. Based on the controversy of the ratings, we analyzed the performance of the machine learning algorithms.

Our results suggest that the effectiveness of machine-learning classification and human-based classification are comparable, yet machines are outperformed by human-based classification. Also, they indicate that classifying narrative contexts is an easier task than classifying valence. Last, results show that machines yield lower accuracy when classifying narrative contexts that are controversial among the human raters.

In addition, we foster scientific reproducibility and continuity by making our dataset available online at `http://www.github.com/aldalmora/NC_Music_Brazil`.

The remainder of this paper is organized as follows. Section 2 describes the methods used in this work. Section 3 discusses the results and their outcome. Section 4 resumes the results and suggests future research about this subject.

2 Methods

In this work, we analyzed the predictive power of text mining tools to classify music lyrics according to their narrative contexts and valence. First, we built a manually-classified ground-truth dataset, as described in Section 2.1. Then, we conducted prediction experiments as discussed in Section 2.2.

2.1 Dataset

The dataset used in this work contains 380 song lyrics extracted from Brazilian Bossa-Nova songs. The lyrics in the dataset were classified according to their narrative contexts. The classification used anonymous raters and an online voting system. Each lyric was classified at least five times. We used the most voted classification as the ground-truth. The narrative context categories used for this classification were *relationships* (songs about a romantic relationships), *emotions or reflections* (songs about abstract impressions on life), and *others* (other themes). Also, the lyrics were classified according to their valence (positive, neutral, or negative) using the same procedure.

The number of songs in each narrative context and valence categories are shown in Table 1. Lyrics that did not have a most voted option (that is, there was a draw between two or more options) within subject or valence were discarded from the dataset.

Table 1: Description for the dataset classified by anonymous rankers showing the amount of lyrics classified according to their narrative context (Relationships, Emotions/Reflections, or Others) and their valence (Positive, Neutral, or Negative).

Context	Pos.	Neu.	Neg.	**Total**
Rel	35	40	29	**138**
Emo/Ref	41	37	10	**123**
Others	10	18	7	**44**
Total	**86**	**95**	**46**	**227**

The next section presents the predictive analysis using text mining and machine-learning algorithms.

2.2 Predictive analysis

In this section, we describe our results related to predictive analysis. We tested several classifier variations, as depicted in Figure 1. Each of the variations used a different representation for the documents.

The first variation regards removal of stopwords. Stopwords are words that are too common in a language and convey little meaning. For this reason, stopwords can be removed from text documents without changing its meaning. However, they can be important for predictive analysis. Hence, we evaluated classifiers both with and without stopword removal.

After deciding for removing or keeping stopwords, each word in the lyrics is mapped to a vector $l \in \mathbb{R}^P$, where P is the total number of words in the whole dataset and l_p is the TF-IDF [18] rating of word p. The whole set of D lyrics vectors are grouped in a matrix $\boldsymbol{L} \in \mathbb{R}^{D \times P}$. Then, the document matrix $\boldsymbol{L}$ is factorized into two other matrices, as:

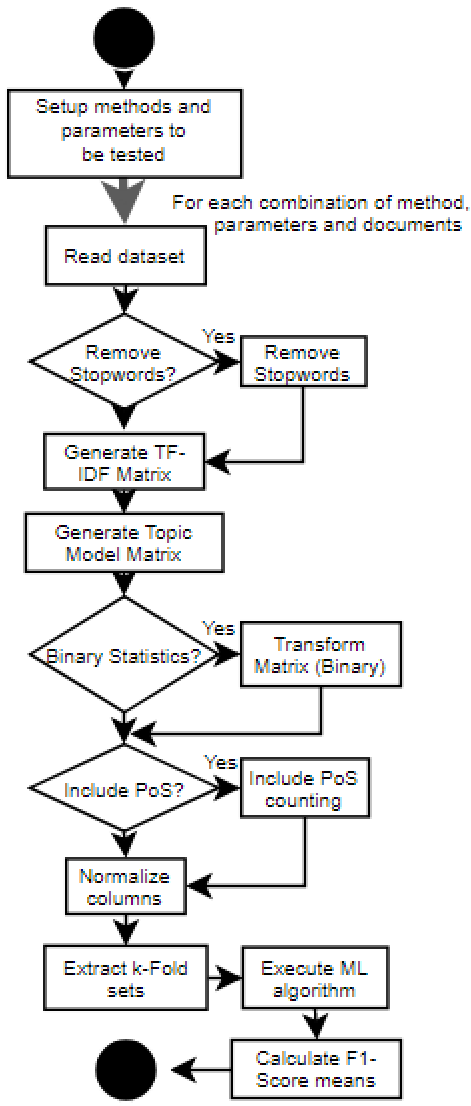


Figure 1: Flowchart depicting the evaluation script for testing diverse configurations and hyper-parameters of the ML algorithms.

$$\boldsymbol{L} \approx \boldsymbol{D}\boldsymbol{W}. \tag{1}$$

In Equation 1, matrix $\boldsymbol{D}$ represents the association of each document to an specific topic, and matrix $\boldsymbol{W}$ represents the association between topics and dataset words. The number of topics n is the inner dimension in the multiplication $\boldsymbol{D}\boldsymbol{W}$, and is a hyperparameter of this process. The values of the elements of $\boldsymbol{D}$ and $\boldsymbol{W}$ are obtained by minimizing:

$$|\boldsymbol{L} - \boldsymbol{D}\boldsymbol{W}|^2 + \lambda|\boldsymbol{D}|, \tag{2}$$

where $\lambda = 0.5$ is a regularization factor used to foster sparsity in D by minimizing the L1-norm $|\boldsymbol{D}|$.

We evaluated two variations of the topic ($\boldsymbol{D}$) representation. The first directly used the results of minimizing Equation 2, meaning the strength of a topic in each lyric. The second one used a binary representation for $\boldsymbol{D}$ in which all non-zero values were mapped to one, meaning the presence or absence of a topic in each lyric.

Another variation regards the use of part-of-speech (PoS) tags. These tags are related to the classification of words as verbs, nouns, and others. They were used by counting the amount of each tag present in each document and including as columns in the representation matrix $\boldsymbol{D}$.

After building the representation matrix of the documents, we used a K-Folds cross-validation schema for testing the algorithm. We used the mean unweighted F1-Score as the evaluation metric. The F1-Score is calculated as:

$$\text{Precision} = \frac{\text{\# true positives}}{\text{\# true positives} + \text{\# false positives}} \quad (3)$$

$$\text{Recall} = \frac{\text{\# true positives}}{\text{\# true positives} + \text{\# false negatives}} \quad (4)$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

We tested Support-Vector Machines (SVM), K Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), and Random Forest (RF) classifiers. All feature set combinations were tested with each of the classifiers.

Additionally, we compared the machine-based classification accuracy to the average human classification accuracy. For such, we used three different subsets of the dataset. They respectively comprised song lyrics whose most voted label had 3, 4, and 5 votes. This separation aimed at detecting whether more controversial (from the raters' perspectives) lyrics is related to different machine-learning based classification accuracy.

The results of these tests are discussed in the next section.

3 Results and Discussion

The results presented in this section are divide in three parts. First we analysis the outcome from machine learning predictions. Then, in the second part, we analysis the consistency of the ground-truth used in for training ML models, focusing on the online raters classifications and their divergence. Last we present and discuss the behavior of the ML algorithms related to controversial lyric classifications by humans.

3.1 Machine-Learning Classification

After executing the different methods described in the previous section, we compiled the best results as presented in Table 2 and Table 3. Topics(from topic analysis) indicates how the topic representation was quantified. PoS indicates whether Part-Of-Speech was considered or not. SW indicates if the stopwords were removed from the lyrics. Table 2 contains results generated with the objective of predicting the context. Table 3, with the objective of predicting the valence. The 0R classifier works by selecting the most frequent class in the dataset and was used as a baseline.

Table 2: Accuracy and per-class F1-Scores for predicting the Context.

Method	Topic	PoS	SW	F1	Acc.
KNN	Binary	No	No	0.49 $\pm$0.14	0.56
RF	Count	No	No	0.44 $\pm$0.06	0.62
SVC	Binary	No	No	0.51 $\pm$0.11	0.56
GNB	Count	No	No	0.52 $\pm$0.07	0.55
0R	-	-	-	-	0.45

Table 3: Accuracy and per-class F1-Scores for predicting the Valence.

Method	Topic	PoS	SW	F1	Acc.
KNN	Binary	Yes	No	0.50$\pm$0.14	0.55
RF	Binary	No	No	0.40$\pm$0.11	0.51
SVC	Binary	Yes	No	0.51$\pm$0.10	0.56
GNB	Binary	No	No	0.51$\pm$0.13	0.52
0R	-	-	-	-	0.42

The results shown in tables 2 and 3 show that removing stop words does not increase classification accuracy. Also, they indicate that Part-of-Speech tags are only relevant for predicting valence. Last, we note that the Gaussian Naive Bayes classifier had the best performance in both cases.

These results were compared to the accuracy of human raters, as shown in the next section.

3.2 Comparison to Human classification

The dataset used in this work was the result of voting using human raters. This means that raters can disagree with each other. The test described in this section evaluates the rating dispersion of each rater.

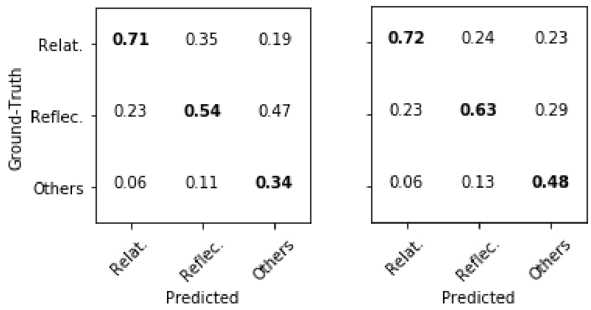
In this test, we consider each rater as a predictor, and calculate their F1-Score. After that, we calculated the average F1-Score for each classification task (predicting context or valence). The results are shown in Table 4.

Table 4: F1-Score for classification performed by humans and machines each dataset rater as a predictor.

Target	0R	Machine	Humans
Context	0.45	0.525$\pm$0.07	0.644$\pm$0.27
Valence	0.42	0.511$\pm$0.13	0.565$\pm$0.28

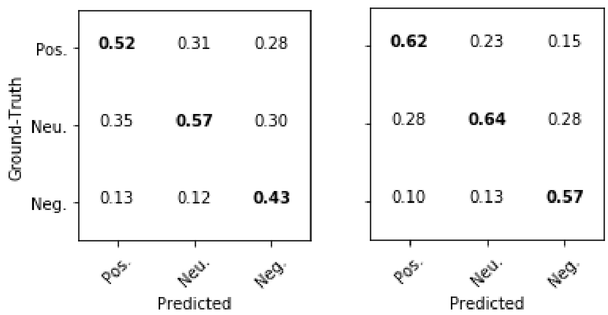
The confusion matrices related to the human and machine classifications are shown in Figures 2 and 3.

We can notice that machine learning was unable to outperform human classification in all objectives and classes. Also, we can notice that the *relationship* class is related to less errors in the narrative context classification.



(a) Machine **(b) Humans**

Figure 2: Confusion matrices for the prediction of narrative contexts.



(a) Machine **(b) Humans**

Figure 3: Confusion matrices for the prediction of valence.

Last, we can see that *neutral* valence was the easiest to classify.

Next, we show how machine learning classifiers behave with more controversial lyrics.

3.3 Behavior with Controversial Lyrics

Some lyrics in our dataset were classified more consistently than others. This consistency can be measured according to the number of human raters that agree with the most frequently voted label. Using this criterion, we can build the following subsets for our dataset:

- $G5$ - Five votes
- $G4$ - Four votes
- $G3$ - Three votes

By grouping the results relate to the best classifier discussed in the previous sections according to these subsets, we obtain the classification accuracy shown in Table 5 for context labelling and Table 6 for valence labeling.

These results show that lyrics that are rated more consistently in narrative contexts can also be classified more accurately by the machine. This suggests that the machine learning algorithms are converging towards learning behaviors that are similar to common sense.

However, this behavior cannot be observed for valence classification. We speculate that valence ratings are more closely related to each rater's personal experiences

Table 5: Accuracy results for narrative context classification divided by lyric rating divergence.

Method	G5	G4	G3
KNN	0.63	0.56	0.49
RF	0.74	0.63	0.53
SVC	0.58	0.60	0.51
GNB	0.69	0.56	0.51
Human	1.00	0.80	0.60

Table 6: Accuracy results for valence classification divided by lyric rating divergence.

Method	G5	G4	G3
KNN	0.44	0.49	0.48
RF	0.50	0.54	0.47
SVC	0.50	0.54	0.52
GNB	0.49	0.49	0.51
Human	1.00	0.80	0.60

than to objective elements of text. This reflected on the lower classification ratings for humans as shown in Table 4, and in a lower consistency in the data yielded to machine-learning algorithms.

The next section presents conclusive remarks.

4 Conclusion

This paper described a series of experiments using text mining tools to classify lyrics based on their narrative contexts and their valences. Our results were compared to the ones yielded by human raters using a dataset built specifically for this work.

The results show that classification results are sensibly different when different target labels are used. Also, they show that different document representations can be more effective for the classification of different labels.

Results also show that valence classification is a harder task than narrative context classification. We speculate that this is because valence rating is highly influenced by the rater's personal experience, whereas rating narrative contexts is a more objective task.

Last, results show that narrative contexts that are more controversial for human raters are also harder to classify using machine learning algorithms. This indicates that, in this case, the classification algorithm behaves similarly to the rater's common sense.

This work did not evaluate representations that take phrase structures into account, like attention networks. This poses an interesting avenue for future work.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001.

References

[1] Thomas Scheff. *What's Love got to do with It? Emotions and Relationships in Popular Songs*. Routledge, 2016.

[2] Yi-Hsuan Yang and Homer H. Chen. Prediction of the distribution of perceived music emotions using discrete samples. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2184 – 2196, 2011.

[3] Andrew H. Gregory. *The Social Psychology of Music*, chapter 7, pages 123–140. Oxford University Press, 1997.

[4] Eduard Hanslick. *On The Musically Beautiful*. Hackett Publishing Company, 1986.

[5] Lisa Feldman Barrett. The theory of constructed emotion: an active inference account of interoception and categorization. *Social Cognitive and Affective Neuroscience*, 12(1):1–23, 10 2016.

[6] Yukiko Uchida, Sarah S. M. Townsend, Hazel Rose Markus, and Hilary B. Bergsieker. Emotions as within or between people? cultural variation in lay theories of emotion expression and inference. *Personality and Social Psychology Bulletin*, 35(11):1427–1439, 2009. PMID: 19745200.

[7] Joseph Campbell. *The hero with a thousand faces*. New World Library, 2008.

[8] C. G. Jung. *The Archetypes and the Collective Unconscious*. Princeton University Press, 1980.

[9] Paula Dornhofer Paro Costa. *Two-Dimensional Expressive Speech Animation*. PhD dissertation, Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação, 2015.

[10] James A Russell and Lisa Feldman Barrett. Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant. *Journal of personality and social psychology*, 76(5):805, 1999.

[11] Sanjeev Arora;Rong;Ankur Moitra. Learning topic models - going beyond SVD. *CoRR*, abs/1204.1956, 2012.

[12] C. Jareanpon, W. Kiatjindarat, T. Polhome, and K. Khongkraphan. Automatic lyrics classification system using text mining technique. In *2018 International Workshop on Advanced Image Technology (IWAIT)*, pages 1–4, Jan 2018.

[13] Y. Hu and M. Ogihara. Identifying accuracy of social tags by using clustering representations of song lyrics. In *2012 11th International Conference on Machine Learning and Applications*, volume 1, pages 582–585, Dec 2012.

[14] Swati Chauhan;Prachi Chauhan. Music mood classification based on lyrical analysis of hindi songs using latent dirichlet allocation. *2016 International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds*, pages 72–76, 2016.

[15] Stuti Shukla;Pooja Khanna;Krishna Kant Agrawal. Review on sentiment analysis on music. *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, pages 777–780, 2017.

[16] Lili Nurliyana Abdullah Teh Chao Ying, Shyamala Doraisamy. Genre and mood classification using lyric features. *2012 International Conference on Information Retrieval & Knowledge Management*, pages 260–263, 2012.

[17] Felipe S. Tanios;Tiago F. Tavares. Impact of genre in the prediction of perceived emotions in music. *16th Brazilian Symposium on Computer Music*, 2017.

[18] J.D. Rajaraman, A.;Ullman. Data mining. In *Mining of Massive Datasets*, pages 1–17. Cambridge University Press, 2011.

Iracema: a Python library for audio content analysis

Tairone N. Magalhães[1]*, **Felippe B. Barros**[1]†, **Mauricio A. Loureiro**[1]

[1]CEGeME - Center for Research on Musical Gesture and Expression – UFMG
Av. Antônio Carlos, 6627 – 31270-010, Belo Horizonte, MG

`tairone@ufmg.br, felippebb@ufmg.br, mauricioloureiro@ufmg.br`

Abstract. *This paper introduces the alpha version of a Python library called Iracema, which aims to provide models for the extraction of meaningful information from recordings of monophonic pieces of music, for purposes of research in music performance. With this objective in mind, we propose an architecture that will provide to users an abstraction level that simplifies the manipulation of different kinds of time series, as well as the extraction of segments from them. In this paper we: (1) introduce some key concepts at the core of the proposed architecture; (2) list the current functionalities of the package; (3) give some examples of the application programming interface; and (4) give some brief examples of audio analysis using the system.*

1 Introduction

Despite the fact that music performance is a central element to nearly every culture, its empirical study is relatively recent, with the seminal works dating back to the turn of the twentieth century. As stated by Clarke in [1], "only once methods had been developed to record either the sounds of performance, or the actions of instruments, was any kind of detailed [empirical] study possible". Over the last few decades, we have witnessed considerable growth in this field of study [2, 3], and the availability of new tools and technologies for extracting information from performances have played a pivotal role in this surge. We believe that the continuous development of more specialized tools to extract information from music performance, as well as better techniques for obtaining more meaningful representations of musical content, will be of crucial importance to continually support empirical research in music performance.

In this scenario, we introduce the alpha version of *Iracema*, a Python package for audio content analysis aimed at the empirical research on music performance. It provides functionalities for extracting patterns of manipulation of duration, energy, and spectral content from monophonic audio, specially for instruments such as clarinet, flute, trumpet, and trombone. Its development was motivated by research projects conducted at CEGeME[1], and was strongly inspired by a previous Matlab tool developed by the group, called Expan [4], which has not been released for public use.

In contrast to instruments like guitar or piano, in which the excitation that produces sound only happens at the beginning of a note[2], in woodwind and brass instruments, the player continuously feeds energy into the system, by means of high pressure air from his lungs. Therefore, due to the dynamic control that the player has over the acoustic properties of the sound, a single note might contain a lot of important expressive information, e.g., timbral manipulations, or dynamic intensity variations. It is harder to extract this kind of information from polyphonic musical signals, such as a full orchestral recording, than from signals of a single source. So a reasonable approach is to use monophonic recordings to better understand them. Another characteristic of the instruments of our interest, is that they can produce very soft attacks, which makes the precise detection of note onsets tricky, especially avoiding the occurrence of false positives. Thus, the techniques implemented on Iracema focus mainly on the extraction of information from monophonic sounds with soft note attack[3].

2 Iracema

Iracema is licensed under the GNU General Public License v3.0, and its source code can be freely obtained at `https://github.com/cegeme/iracema`. To obtain more detailed information about the library, like usage examples, more information about the feature extractors available, library modules, and extensive documentation of the API, check the online documentation, which is available at `https://cegeme.github.io/iracema`.

Iracema uses NumPy arrays for storing and manipulating data, providing a new level of abstraction on top of such objects [5]. It also wraps some functionalities from SciPy [6] to provide methods with a more natural interface for audio content extraction operations, as well as compatibility with Iracema's objects.

2.1 Architecture

Software architecture refers to the set of structures needed to reason about a system. These structures are comprised of software elements, relations among them, and properties of both elements and relations [7]. This section will discuss some import aspects of Iracema's architecture and offer an overview of the elements that compose the core functionalities of the library.

Audio content analysis systems rely on the manipulation of dynamic data, i.e., data that represent an attribute's changes over time. Thus, *time series* is a fundamental element in Iracema's architecture. The starting

*Supported by CAPES.
†Supported by CNPq.
[1]http://musica.ufmg.br/cegeme/

[2]I.e., the plucking of strings in a guitar or a hammer hitting the strings of a piano.

[3]Monophonic sounds with soft note attacks motivated the development of the system, but the reader should be aware that some functionalities of Iracema could also be applied to polyphonic or percussive sounds.

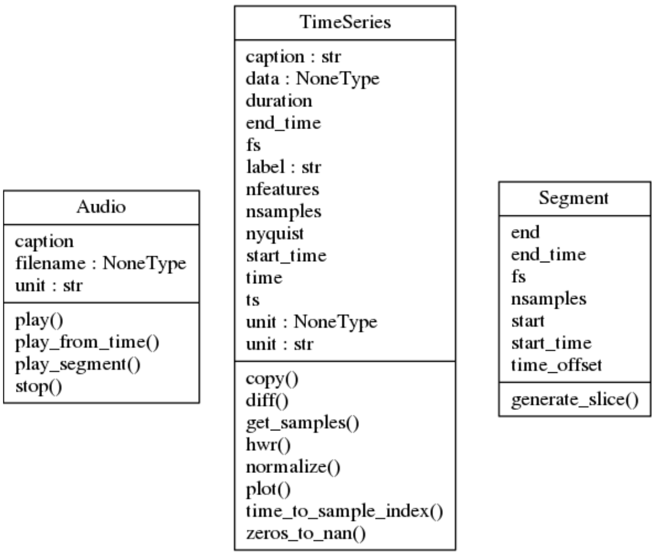


Figure 1: Diagram showing the core classes of Iracema.

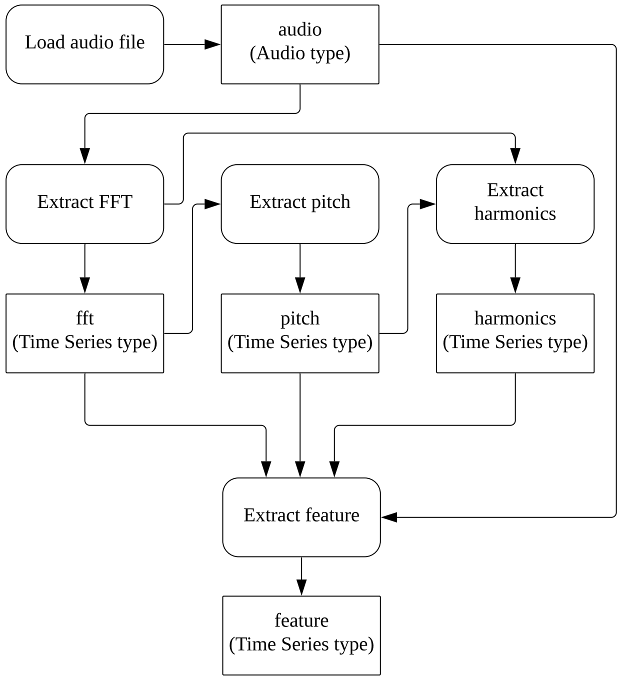


Figure 2: Extracting features from an audio file.

point for any task performed by the system is the *audio time series*, from which other kinds of time-related data will be extracted. The transformation of time series into other time series, to obtain more meaningful representations of the underlying audio, is a common behavior of audio content analysis systems, usually called *feature extraction*. The implementation of such extractors usually depends on some recurrent types of operations, like applying sliding windows to a series of data, for example. In Iracema, these operations are called *aggregation* methods.

Sometimes it will be necessary to deal with a specific excerpt of a time series, such as a musical phrase or a note. There is another important element in the architecture, called *segment*, that can be used to delimit such excerpts. A user may sometimes specify the limits for a segment, within the time series, if he is already aware of its beginning and end; however, most of the time, users will expect the system to identify such limits by itself, a common kind of task in audio content extraction, known as *segmentation*.

Some of the aforementioned elements, like audio, time series, and segments have been implemented as classes, since they have intrinsic data (e.g., the samples of the time series, and the start/end of the segments) and behaviour (e.g., generating time vectors in time series or calculating indexes in segments). Figure 1 shows those classes in a diagram. The `Audio` class inherits the functionalities from `TimeSeries`, and add some specific behaviours (such as loading wave files). `Segments` provide a handy way to extract corresponding excerpts from time series of different sampling rates, since it performs all the necessary index conversion operations to extract data that coincide with the same time interval.

Other elements have been implemented as methods that take objects of those classes as input and output another object. For example, the method `fft` takes as input an `audio` object, a `window_size`, and a `hop_size`, and generates a time series in which each sample contains all the bins of the FFT for the interval corresponding to `hop_size`. Another example, the method `spectral_flux` will take a time series containing the result of an FFT operation as input and generate another time series containing the calculated spectral flux. Figure 2 shows a diagram that illustrates the typical workflow for performing basic feature extraction from audio files.

Segmentation methods will usually take `time_series` objects as input to output a list of segments (Figure 3). Then, these segments can be used to easily extract excerpts from time series objects (Figure 4), using square brackets (the same operator used in Python to perform indexing/slicing operations).

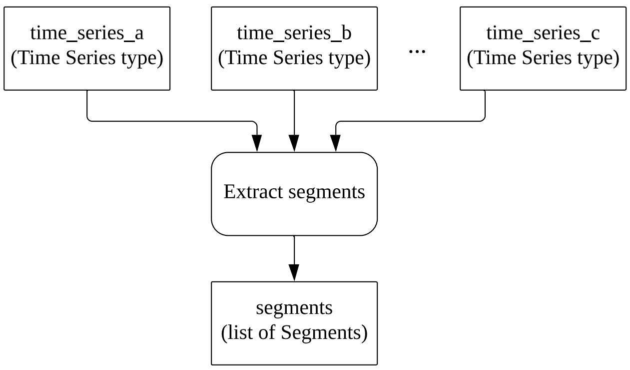


Figure 3: Extracting segments from time series.

2.2 Modules and functionalities

These are the modules that compose Iracema, and their respective functionalities:

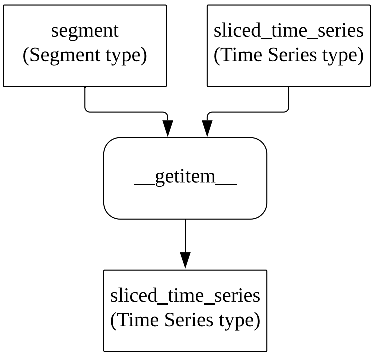


Figure 4: Using a segment to slice a time series.

- `timeseries`: contains the definition of the classes `TimeSeries` and `Audio`.
- `segment`: contains the definition of the classes `Segment` and `SegmentList`.
- `spectral`: contains methods for frequency domain analysis (currently the FFT);
- `pitch`: a few different models for pitch detection.
- `harmonics`: a model for extracting harmonic components from audio.
- `features`: contains methods with the implementation of several classic feature extractors.
- `segmentation`: methods for automatic audio segmentation.
- `plot`: contains several different methods for plotting time series data.
- `aggregation`: contains some common aggregation methods that can be useful for implementing feature extractors.
- `io`: subpackage containing IO methods, for loading/writing files, playing audio, etc.
- `util`: subpackage containing some useful modules for unit conversion, DSP, windowing operations,etc.

2.3 Pitch detection models

The module `pitch` contains models for pitch detection. At the time this paper was finished, two methods had been implemented, as well as an extra method that wraps a model from an external library. The following list shows the pitch methods available:

Harmonic Product Spectrum Measures the maximum coincidence for harmonics, based on successive down-sampling operations on the frequency spectrum of the signal [8].

Expan Pitch Based on the algorithm implemented in Expan [4]. It chooses the highest peaks in the frequency spectrum of a signal as potential candidates, and then extract their theoretical harmonics. The candidate with the higher harmonic energy is chosen as the fundamental frequency.

CREPE Based on a deep convolutional neural network operating directly on the time-domain waveform input [9]. This is a wrapper that uses an external library.

2.4 Feature extractors

These are the methods available in the module `features`:

Peak Envelope Extracts the envelope of the waveform by extracting the peaks in the amplitude for each analysis window.

RMS Calculate the root mean square of a time-series. This is usually a good better choice for extracting the envelope of an audio signal, since it is more closely related to our perception of intensity that the peak envelop.

Zero-crossing The zero crossing is a measure of how many time-series a signal crosses the zero axis in one second. It gives some insight on the noisiness character of a sound.

Spectral Flatness Gives an estimation of the noisiness/sinusoidality of an audio signal. It might be used to determine voiced/unvoiced parts of a signal [10].

HFC Measures of the amount of high frequency content of a time-series spectrum. It produces sharp peaks during attacks transients [11] and might be a good choice for detecting onsets in percussive sounds.

Spectral Centroid The spectral centroid is a well known timbral feature that is used to describe the brightness of a sound. It represents the center of gravity of the frequency components of a signal [12].

Spectral Spread Gives an estimation of the spread of the spectral energy around the spectral centroid [10].

Spectral Flux : Measures the amount of change between adjacent spectral frames [13].

Spectral Skewness Measures how symmetric is the distribution of the values for the spectral magnitudes around their arithmetic mean [14].

Spectral Kurtosis Measures if the distribution of the spectral magnitude values is shaped like a Gaussian distribution or not [14].

Spectral Rolloff The spectral rolloff is a measure of the bandwidth of the spectrum [14]. It is defined as the point in the spectrum bellow which a percentage k of the spectral energy is contained.

Spectral Entropy Measures the unpredictability or disorder in the distribution of the spectral energy [15].

Spectral Energy The total energy of a frame of the spectrum.

Harmonic Energy The total energy of the harmonic partials of a time-series.

Inharmonicity Determines the divergence of the time-series spectral components from an ideal harmonic spectrum.

Noisiness It is the ratio of the noise energy to the total energy of a signal. Represents how noisy a signal is (values closer to 1), as oposed to harmonic (values close to 0) [10].

Odd-to-Even Ratio It is the ratio between the energy of the odd and even energy harmonics of a signal.

3 Examples

This section shows some basic code examples for the library.

```
1 import iracema
2
3 # loading audio file
4 audio = iracema.Audio(
5   "audio/03 - Clarinet - Fast Excerpt.wav")
6
7 # plotting waveform
8 audio.plot()
9
10 # playing audio
11 audio.play()
```

Listing 1: Loading audio and plotting waveform.

Loading audio files in Iracema is pretty straightforward, and the only thing that must be specified is a string containing the path to the audio file that should be loaded[4]. In code listed above, the initializer method for the class `iracema.Audio` shown in line 4 will load the content of the wave file into an `audio` object. Then, the object's `plot()` method (line 8) will display its waveform (shown in Figure 5), automatically setting some basic plot parameters, such as axis labels and title, by using metadata from the audio time series. In line 11, the method `play()` will reproduce the corresponding audio.

```
12 # calculate FFT
13 window_size, hop_size = 2048, 1024
14 fft = iracema.spectral.fft(
15     audio, window_size, hop_size)
16
17 # plot spectrogram
18 iracema.plot.plot_spectrogram(
19     fft, fftlim=(0,10000))
```

Listing 2: Calculating FFT and plotting spectrogram.

The method `iracema.spectral.fft()` shown in line 14 will calculate the FFT for the audio file, using a sliding window of size 2048, with 1024 samples of overlap. It will generate another time series object as output, which will contain multiple values per sample, one corresponding to each bin of the FFT. It will then be passed to the method `iracema.plot.plot_spectrogram()` in line 18, to obtain the visualization shown in Figure 6.

```
20 # extract the pitch and then the harmonics
21 pitch = iracema.pitch.hps(fft)
22 harmonics = iracema.harmonics.extract(fft, pitch)
23
24 # plot the harmonics over the spectrogram
25 iracema.plot.plot_spectrogram_harmonics(
26     fft, pitch, harmonics['frequency'],
27     fftlim=(0,12000))
```

Listing 3: Extracting pitch and harmonics.

The pitch will be extracted from the signal using the method `iracema.pitch.hps()` in line

[4]Iracema uses the library audioread [16] to load audio files, and can handle different audio file formats.

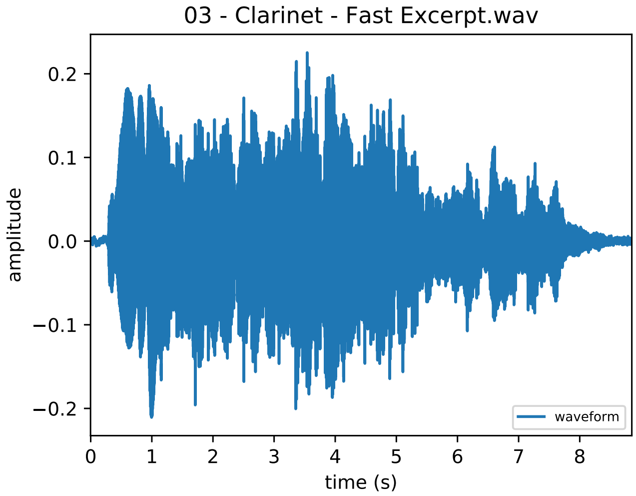


Figure 5: Waveform for the audio file loaded.

21, and then, in the next line, passed as a parameter to the method `iracema.harmonics.extract()`, along with the previously calculated FFT. The method for extraction of harmonics will extract 16 harmonics by default, but a different number could be specified, using the optional argument `nharm`. In line 25, both objects will be plotted over a spectrogram using the method `iracema.plot.plot_spectrogram_harmonics()`, resulting in the plot show in Figure 7.

```
28 # extract some features
29 sflux = iracema.features.spectral_flux(fft)
30 sflat = iracema.features.spectral_flatness(fft)
31 sc = iracema.features.spectral_centroid(fft)
32 no = iracema.features.noisiness(fft,
33     harmonics['magnitude'])
34 hfc = iracema.features.hfc(fft)
35
36 # plot waveform and other features
37 iracema.plot.plot_waveform_trio_and_features(
38     audio, features=(sflux, sflat, sc, no, hfc))
```

Listing 4: Feature extraction.

In lines 29-34, five different features will be calculated for the whole audio signal. Then, they will all be plotted (line 37), along with a visualization of the waveform, using the method `iracema.plot.plot_waveform_trio_and_features()` (Figure 8).

4 Future perspectives

The functionalities of the library will move towards feature extractors that can provide more meaningful representations of the information from music performance, from a musical point of view. The architecture proposed for Iracema and the feature extractors mentioned in this article form the basis for the development of such representations, which will be included in the future stages of development of the tool.

Good models for note segmentation are essential for audio content extraction, so this is our major concern

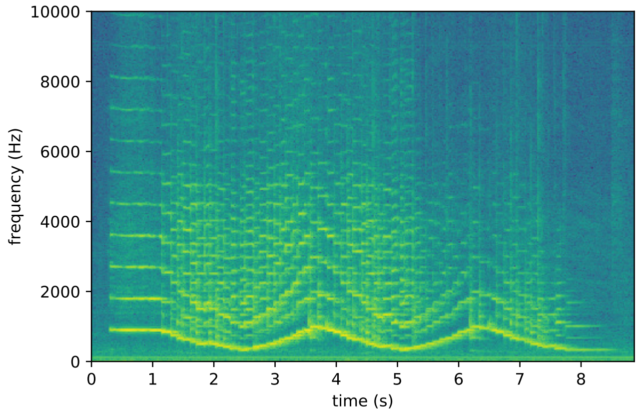


Figure 6: Spectrogram for the audio file loaded.

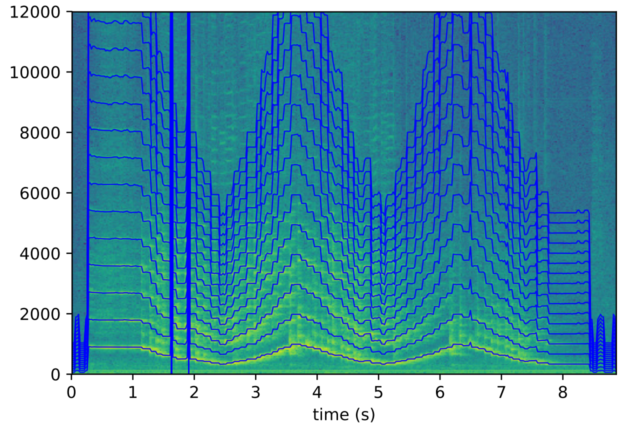


Figure 7: Frequency for the harmonics components extracted (the plot has been limited to the range 0-12 KHz using the parameter *fftlim*).

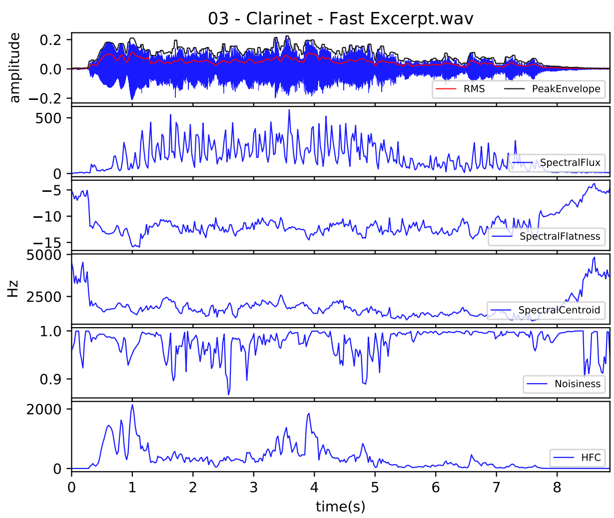


Figure 8: Five different features exctracted for the audio file loaded.

at the time. Although we have already implemented some basic models that use pitch and energy information to detect note onsets, sometimes they produce some false positives, therefore, we are working on a better model, using machine learning techniques, which should be included in a future release. Such robust note segmentation is essential for obtaining good articulation descriptors, and we have already developed a legato index descriptor that relies on such robustness. We also plan to include, in a future version of Iracema, a vibrato descriptor, which was previously proposed and described in [17].

Acknowledgement

The development of this work has been supported by CAPES/Brazil (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and CNPq/Brazil (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

References

[1] Eric Clarke. Empirical methods in the study of performance. In *Empirical musicology: Aims, methods, prospects*, pages 77–102. 2004.

[2] Alf Gabrielsson. Music performance research at the millennium. *Psychology of Music*, 31(3):221–272, 2003.

[3] C Palmer. Music performance. *Annual review of psychology*, 48:115–38, 1997.

[4] Campolina, Thiago A. M. and Mota, Davi A. and Loureiro, Mauricio A. Expan: a tool for musical expressiveness analysis. In *Proceedings of the 2nd International Conference of Students of Systematic Musicology*, pages 24–27, 2009.

[5] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[6] Eric Jones, Travis Oliphant, Pearu Peterson, and Others. SciPy: Open source scientific tools for Python, 2001.

[7] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Publishing Company, third edit edition, 2013.

[8] P De La Cuadra. Efficient pitch detection techniques for interactive music. In *ICMC*, pages 403–406, 2001.

[9] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. CREPE: A Convolutional Representation for Pitch Estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018*, 2018.

[10] G. Peeters, B.L. Giordano, P. Susini, N. Misdariis, and S. McAdams. The timbre toolbox: extracting audio descriptors from musical signals. 130(5), 2011.

[11] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1046, 2005.

[12] Tae Hong Park. *Introduction to digital signal processing: Computer musically speaking*. World Scientific Publishing Co. Pte. Ltd., 2010.

[13] Simon Dixon. Onset Detection Revisited. In *9th International Conference on Digital Audio Effects*, pages 133–137, Montreal, Canada, 2006.

[14] Alexander Lerch. *An introduction to audio content analysis: Applications in signal processing and music informatics*. 2012.

[15] Aik Ming Toh, Roberto Togneri, and Sven Nordholm. Spectral entropy as speech features for speech recognition. *Computer Engineering*, (1):22–25, 2005.

[16] Adrian Sampson. Audioread. `https://github.com/beetbox/audioread`, 2011.

[17] Magalhães, Tairone N. and Mota, Davi A. and Neto, Aluizio B. O. and Loureiro, Mauricio A. Análise do vibrato e bending na guitarra elétrica. In *Anais do XV Simpósio Brasileiro de Computação Musical*, pages 36–47, 2015.

A technical approach of the audience participation in the performance "O Chaos das 5"

João Teixeira Araújo, Avner Maximiliano de Paulo, Igino de Oliveira Silva Junior, Flávio Luiz Schiavoni, Mauro César Fachina Canito, Rômulo Augusto Vieira Costa

[1] Arts Lab in Interfaces, Computers, and Everything Else - ALICE
Federal University of São João del-Rei – UFSJ
São João del-Rei - MG - Brazil

teixeira.araujo@gmail.com, avnerpaulo.mg@gmail.com

igino@ufsj.edu.br, fls@ufsj.edu.br

mauro.cesar.10@hotmail.com, romulo_vieira96@yahoo.com.br

Abstract. *Since HTML 5 and web audio were released, we have seen several initiatives to construct web based instruments and musical applications based on this technology. Web based instruments involved composers, musicians and the audience in musical performances based in the fact that a web instrument embedded in a web page can be accessed by everyone. Nonetheless, despite the fact that these applications are accessible by the network, it is not easy to use the network and these technologies to synchronize the participants of a musical performance and control the level of interaction in a collaborative musical creation scenario. Based on a multimedia performance created in our research group, O Chaos das 5, we present in this paper some scenarios of interaction and control between musicians and the audience that can be reached using a server side programming infrastructure along with the HTML5. In this performance, the audience took part of the musical soundscape using a cellphone to access a set of digital instruments. These scenarios and the proposed solutions brought up a set of possibilities to balance control and interaction of audience participation into live performance using web instruments.*

1 Introduction

Formerly, the music of European tradition had 3 well-defined roles: the composer responsible for creating the music, the performer responsible for playing it, and the audience, which until then had the role of just prestiging it and clap the hands in the end. However, different approaches and techniques were being created over time to break these roles so that the public became more and more engaged in an artistic performance.

About the 1950s, Allan Kaprow came up with a concept called Happenings, which were "spontaneous theatrical events with some element of spontaneity or improvisation that never repeats in the same way with each new performance". As an example of these Happenings, John Cage wrote a piece called 4'33, first performed at Woodstock, New York, in 1952, which began to make a conceptual break / change in the way music is produced [1] according to the audience participation. In this work, the main performer takes the stage, opens his piano and remains silent for four and a half minutes, consequently the audience begins to make certain noises and this is the aesthetic result expected by the artist. Thus, in 4'33, the public began to receive a more active role, being able to modify the result of the work in question, acting as a performer and getting an unique experience each time the work was performed.

With technological advancement, new types of artistic performances with the participation of the public were emerging, such as the so-called digital performances, which are performances mediated by some type of technology, as cellphones, sensors, projectors, among others. Benjamin Taylor affirms that the art based on cellphones was created as early in the first years of the 21st century [2]. The author studies specifically the technological spectacle Dialtones (2001) as proof of concept of the musical potentials of the new technology, highlighting some motivations expressed by the composers from that time: the ubiquity present in mobile phones, the intention to illustrate wireless in a social space, and the intention to use cell phones as a means of creating art.

The cellphone can also be used to make the audience part of the performance, playing with it and breaking the roles in contemporary music. There are several works that uses mobile devices as the audience interaction device, yet a variety of design choices must still be made when creating a mobile-based audience participation experience [3]. Some works found in the literature will be presented in Section 2.

Following this lead, we present a performance created by the authors called "O Chaos das 5". This participatory performance, presented in Section 3, consists on a musical, visual and gestural dialogue through technology influencing the experience of the audience and being influenced by the interaction of the audience members through mobile devices. Besides that, the technology behind the scenes of our performance is presented in Section 4 with a discussion about the choices and lessons learned. At the end, this paper presents some considerations in Section 5.

2 Related Works

Several works present the participation of the audience in musical performances. The audience using cellphones as instruments in a chamber music work is presented in *12* composed by Radius Ensemble [3]. This work aims to give to audience an “engaging, individualized and influential role in live music performance” but limited the number of participating audience members to grant direct music making and technical transparency.

Another initiative to promote interaction between the audience via smartphones is a DMI’s, called SWARMED [4]. In this work, it is worth mentioning specifically the technical aspects that the authors used for the application of DMI’s created to the public. The authors used a local network from a popular Wi-Fi router, which was in charge of directing all traffic to a laptop, as well as having a captive-portal to redirect users to the application. The used laptop acted as an Internet gateway and was also in charge of acting as a DNS, rewriting all the DNS requests back to the laptop itself, as well as acting as a DHCP server, assigning IP’s to all participants. In all, seven distinct DMIs were developed and the system operated under an average of 20 smartphones (even though it was tested for more than 60 smartphones simultaneously). In general, the only problem encountered was that many users were not aware of how to connect to the supplied network, becoming a drawback that could be handled with simple instructions printed on a flyer.

Another performance used mobile phones not only to the audience. Aiming for high-scale musical performances and using mobiles as DMIs, a musical performance called Echobo [5] was developed, which was based on the use of two types of instruments implemented in a smartphone application: one for the maestro and one for the public. The conductor was responsible for defining the progression of chords, which although it does not emit any sound, it is responsible for controlling the harmony of the sounds produced by the public. The public, from the harmonic constraint set by the conductor, can play note by note within the defined harmonic field. The audience studied was between 20 and 120 participants and the feedback received was that when using the Echobo they felt more connected to music and other musicians. That is, the creation of instruments with user-friendly interfaces seems to be a very promising idea regarding public satisfaction.

Figure 1: The audience participation during the performance O Chaos das 5

If the usage of cellphones as spreaded loudspeakers was a good choice, another option is to to use a server to process audio controlled by the audience in real time through cellphones, like the performance and application presented in [6]. In this performance, the audience could change some audio engine parameters in real time and a central server generated the sound output to the loudspeakers.

Another performance used cellphones to reach the audience participation but without a DMI developed to it. TweetDreams [7] is a performance based on the tweets of the participants collected during a musical performance. These tweets were grouped into graphs of related tweets and associated with melodies. This type of approach ends up generating music according to the network of relationships that a tweet in particular has with other tweets, which escapes the idea of just creating music in the lexical sense of each tweet.

Cellphones can also be used to get audience opinion and use this opinion to decide musical aspects of a performance. A web application to enable the contribution/collaboration between the participants is presented in [8] . The developed application was based on a client/server architecture that allows bidirectional communication for the creation of content. The public, through a voting system, opts for certain predetermined musical attributes that are handled by the musicians and generates a certain type of sound for the audience. Among the studied aspects, the public was very satisfied with the use of the application in general, but dissatisfied with the influence they caused among themselves.

There are performances with a limited number of audience participants and also performances focused on a large number of participants. Based on a cloud server and web audio, [9] developed a digital musical instrument in network that seems to be effective in relation to its scalability. The work is based on a three-way connection, being they human, musical instrument and cloud server. Thus, an interface was built from a web page focused on collaborative music, where the instrument is responsible for the communication with the server, allowing social interaction between the audience, from the sending of messages.

Some initiatives also escaped from the use of cellphone and used other technologies to create audience’s participation. Trying to get away from the idea of listening to music in a unidirectional way, such as the electronic music consumed in club’s where it is played from the DJ, [10] developed a work called Experio. The Experiment allows the audience to dance under various designated areas, marked and illuminated by laser beams. In this way, the laser is constantly interrupted in several areas, thus modifying the sound produced by the musical moderator, which uses an interface to control certain parameters. The work has three main objectives: to allow the participation of even those with limited musical abilities; allow for exploration, expressiveness and improvisation; encourage the creation between the moderator and the audience.

Another form of public/performer interaction that run away from the idea of using the cellphone as a musical instrument for the public is presented in [11]. This work presents a bidirectional communication system in which the performer is in charge of playing a Smart Musical Instrument (SMIs), called Smart Mandolin, which influences and is influenced directly by the audience. Audience uses Musical Haptic Wearables (MHWs) in their arms, producing sounds from a Pure Data application that synthesizes tactile stimuli through pulse width modulation techniques.

Another related work presents a study to understand the behavior of the public in relation to their technical and musical familiarity with the use of a specific DMI in a musical performance [12]. Through the developed work, it is possible that the level of interest of the public in a performance, in general, is not affected according to the knowledge of each participant about the used DMI. In other words, explaining the technical aspects of a DMI for all participants in a performance can be an exhausting and unnecessary action.

3 The Performance “Chaos das 5”

The “Chaos das 5” is an audiovisual digital performance developed by the Transdisciplinary research Group (GTRANS) from the Federal University of São João del-Rei involving the ALICE (Arts Lab in Interfaces, Computers and Else) / Orchidea (Orchestra of Ideas) group from the Computer Science Department and the ECOLAB / Movère group from the Scenic Arts Department.

Figure 2: The performers interacting with their own projection in the final part of the performance.

The guideline of the performance took Alice, from Lewis Carroll book - Alice in the Wonderland [13], as a metaphor to take audience to a synthetic and disruptive wonder world[1]. The audience members could take part of the show and there was no stage or another mark to limit the space of the performers and the audience.

At least three layers of information were used to create an immersive experience to the audience combining music, visual and gesture by the means of technology. Musicians located around the space create the base of the sound using DMIs and over processed electric guitars. The audience members could participate of the sound using their cellphones and accessing a set of web DMI developed to this performance.

The visual layer used projectors, cameras and some software developed in our lab to create images in real time. The code of the software sometimes was also projected, like in a live coding performance, and two programmers were changing the code and the visual on the fly. Images took by the audience members with their cellphones was also used in this layer and webcams and image processing in real time completed the visual set up.

The gestural layer was performed by performers interacting physically with the audience members in the space. In the beginning of the show it was probably not easy to identify who were the artists and who were audience members. This performatic artists merged gestural score and improvisation to interact among them and with audience members and their participation became more clear during the show.

3.1 The plot

We started the performance with the projection of a countdown clock and an invitation to take part of the performance accessing a website (Figure 3). In the website, the audience members could find some instructions and instruments that could be used during the show. These information was hidden and in the beginning of the performance we have a game like a treasure hunting to find some clues to access a key and take part of our experiment. Audience members could also inform their names and upload a picture to register their participation in the performance.

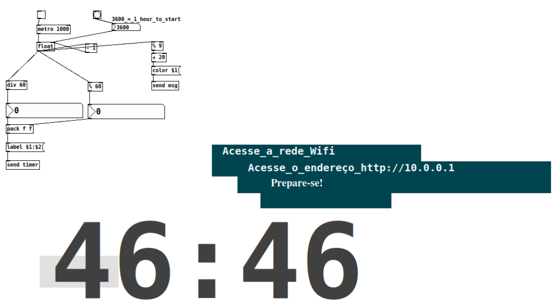


Figure 3: The countdown patch used in the beginning of the presentation.

Finished our treasure hunting, the first part of the

[1]The name of the performance, “Chaos das 5” sounds like a pun since “tea” in Portuguese is “chá”. A translation to English like “5 o’clock chaos” or “5 o’clock tea” do not keep the pun an lost the cacophonic meaning.

performance starts in a synthetic universe, the dive in the rabbit hole, among synthetic images and infinite glissandos that remembers Metastasis from Xenakis. During this part, the performers that were among the audience members started acting in a reverse form, revelling their selves as part of the performance and taking attention to them.

The free falling finishes in a second part, a disruptive experience in the real world. A territory battle in the city where people try to exist and register this existence guided this part of the performance. We projected a noisy sequence of pictures of graffiti, like in Figure 4, and other urban scenes while a city soundscape completed a saturated urban scene.

Figure 4: An urban image projected on the second part of the performance.

Performers started painting their selves using stickers and brushes and the audience members were invited to do the same. This part is the most saturated part of the performance and probably the tensest part of it. Maybe a murder can happen during this part and maybe the pat down can take some audience members.

To escape the reality and the tension of the second part, a third part takes the audience to a surreal experience, calming down until the end. The performers, tired of the second scene, start a slow dance in front of the distorted projection of themselves.

At the end, like a credit film, a projection presents the name of all members of the performance including the audience members that filled their name and picture in the website.

4 The technology behind the scene

To create the participation of the audience it was necessary to provide a small network infrastructure. A local web server and a wireless access point were available to be accessed by the audience members so that no Internet connection was necessary to take part of the concert. A DNS server was also configured to access the website using a name instead of an IP and port.

The website front page asked the audience members' name and gave a few clues to find the key and then they could access a page to select different instruments (Figure 5).

Figure 5: The website's front page.

Each part of the performance had different type of instruments. For the first part, a free falling in the rabbit hole, we developed an application to generate glissando inspired in the Shepard Tone and other instruments with simple sounds, like sinusoids and saw tooth waves, to complete the synthetic scene. The glissando was controlled by a button to start it and the synthetic simple sounds were controlled by the cellphone's accelerometer. Thus, part of the audience should not move and only press a button to play while other part would be using gestures do create sounds.

The second part, the disruptive reality, used a soundscape composed to the piece. The audience had a web sample based instrument to complete the soundscape. Sirens, cars, sprays, traffic jam, church bells (yes, they are common and maybe a signature of our city soundscape) and other sounds could be used to play while the performers were painting themselves. These sounds were available in a web interface just needing to push a button to be started.

The third part, escaping reality, used long sounds to create a peaceful atmosphere. The audience here should stop playing and just relax until the end of the performance.

4.1 The first development: simple Javascript instruments

In the beginning we were thinking about to use a Java Native application to Android to create the DMIs for the audience. Asking to our group members we discovered that nobody has enough space to install a new application in their device. Besides that, there are no compatible language to develop an application to Android and iOS and it would be necessary to create both applications to attend both audience demand. Thus, we chose to develop the instruments using HTML5 technology. As our system does not rely on much processing, we chose to use the Apache web server on a notebook, which is able to meet our demands. At this time we were not thinking about take names and pictures from the audience to use it in the final credits.

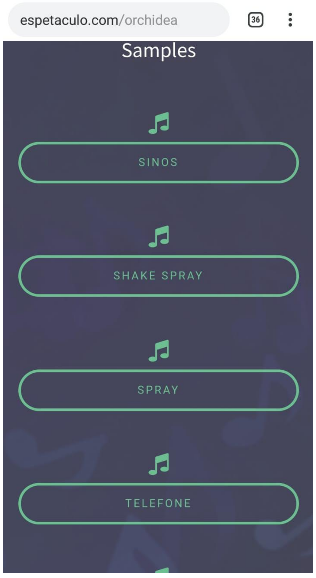


Figure 6: Interface - Samples

For the audience participation we developed 3 set of webaudio instruments, one for each part, using Mosaicode, a visual programming environment developed in our lab.

Specifically about the glissandos instrument, the participant can select the respective volume and density of notes to be played. Altogether 3 types of glissandos were implemented which vary in relation to the range of frequencies to be played, their respective duration and if it is ascending or descending.

The synthetic simple sounds instruments were implemented using a full frequency spectrum being controlled exclusively by the accelerometer. Our intention was to have different voices of simple sounds but the random choice of frequencies did not help to achieve this goal. Thus, this instrument could also have global control, like a conductor, selecting different frequencies to each audience member.

About the sample-based instrument, we chose to select some samples from an urban scene, such as car sounds, horns and bells. It is noteworthy that it was necessary a preprocessing of the samples used to adjust their respective volumes and normalizing their gains.

Unfortunately, we chose to leave the use of instruments free to the public, that negatively influenced our control over the sound layer in each scene of our performance. It would be nice to have a network metronome to synchronize all the instruments according to a global clock. Thus, each glissando, for example, could have a different duration time based on the start frequency and the global clock giving to the performance a sense of synchronization.

The same could happen with the sample based instrument. It would be nice to have all the selected sounds being played during the performance. Giving to the audience members the possibility to choose any sample resulted in lots of devices playing the same samples and no one playing others. A global control could help it too.

Also, it would be nice to provide all instruments before the concert, leaving the audience try it out, but limit the access of the instruments during the performance to avoid unexpected sounds from one part into other parts. Furthermore, to silence all the instruments to the third part could be really necessary.

4.2 The Second attempt: Adding server side programming

To try to give to audience members a possibility to take part of the performance but also to allow us to have control of their part, we decided to use server-side programming to help the implementation of the web DMIs. Two different approaches could be used to synchronize the server and the Javascript clients: the Websocket API or the AJAX HTTP request.

Websocket is a Javascript client-side API available with the HTML5 that creates a network socket for client-server connection enabling a full duplex and low latency communication. An AJAX request is an asynchronous HTTP request that can be used to client-server communication to request an information without reloading the full page. HTTP request is not real time or low latency and can be scheduled to be processed from time to time.

Since our performance did not need an accurate time synchronization, we choose to use HTTP request. Also, HTTP request can be used with a simple server side programming language, like PHP, ASP or JSP and a common webserver to provide the application. We used Java programming language for web application, from the servlet-based JSP technology, with an Apache Tomcat server providing the web instruments. With server side programming, it was easy to reach the desired control and also to create a form to get the audience members name and picture to show it on the final credits.

A second interface, to control all instruments and groups, were implemented to be "the conductor". Using the conductor interface, available in a special link, it was possible to enable or disable instruments, to check connected users, to list names and pictures and to time synch sound events. Thus, from time to time all the instruments from the audience members used to receive time stamp synch messages and also ask the server which instrument was available to be played.

4.3 Under the hood of the performance: lessons learned

The idea of local infrastructure, inspired by the work of [4], with a web server, a wireless access point and a DNS server was really good to grant that everything would work in the presentation and it worked but, of course, we have lessons learned to share.

Different from pieces like *12* [3], our piece had no stage and we did not limit the number of audience members participation. Since our performance happened in a open space, it was impossible to previously known the number of participants and it was totally impossible to scale the necessary infrastructure. A problem about it is that a wireless router has a limited number of connected clients and in a cheap router it is very limited. This information is normally not provided in the manual and it is not easy to test up the performance limitation of a network equipment. Thus, to use a cheap and simple network equipment can be a problem if the number of clients connected to the network increases too much. Also, if the device reaches the limit of connected clients, other clients trying to connect keep sending messages to the device, messing up the network up and creating a really unstable situation. This is our first learned lesson: network devices have limitations and it is important to know them before.

When our connection limit was reached, some audience members tried to use their 3G connection to access an instrument. To make it easy to read and to type we decided to not use a simple string and not a real website URL. Thus, the instrument was not available for all those participants who tried to use their 3G connection to access the instruments. Also, at the end of the presentation, several participants were asking to access the server to play up with the instruments after the show. Another lesson learned: We should had used our real website URL to provide the instruments, even in a LAN set up, to allow users to connect it after the performance.

An unexpected behavior happened due to our DNS configuration. We set up the DNS server to resolve any address in our server. Thus, even if a participant wrote the wrong address it was possible to access our web server and the instruments. It seemed to be a good idea but we did not realized that all the network traffic of the participant devices was redirected to our server. The participants tried out to use the Internet and applications like Netflix, Youtube, Whatsapp, Facebook, Android updates, everything, being responded by our application server that should only provide an instrument. It created a huge waste of processing and a huge bottle neck to real request that brought lots of instability to our system. The lesson learned here is to set up a DNS only to the correct address to avoid extra network traffic.

Figure 7: Before the performance starts, audience members can explore and try out the available instruments

Another lesson was learned about the space utilization. We did not want to define where the audience would be and where the performance would happen. Indeed, we would like to have people every place and to perform among the audience. All we had defined were where to place musicians, projectors and live coders with laptops to the live projections. When the performance began and we turned the projectors on, this arrangement ended up defining the audience space. Obviously, people avoided to be in front of the projectors lights and in a few time we had the undesired borders between the public space and the performance space. Maybe it is impossible to convince the participants to be in front of the projectors.

In addition to the various lessons learned from the show and in relation to the other related works studied, it is worth emphasizing four aspects that should be taken into account when developing a performance concerned with public satisfaction: when creating a DMI, the interface used should be simple and user-friendly so the audience is satisfied [5]; the use of a captive portal facilitates and simplifies the process of redirecting the public to the created DMI [4]; the interaction of the public with the performance, from votes capable of modifying sound aspects, seems not to be a good idea in relation to public satisfaction [5] and finally, in a performance, the technical explanation of DMI used can be an exhausting and unnecessary action [12].

About how to use mobiles to engage the audience in the performance, this experience give us some clues of what else can be explored in future presentations. We noticed that we could also explore the cellphone lights changing the background color of our web application to create a penumbral illumination of the scenes, specially at the third part of the performance. It could be interesting also to use the cellphone flash to create a strobe light effect with the help of the audience. The vibracall could be used to emphasize the button click giving haptic feedback to the instrument.

5 Final considerations

The Chaos das 5 performance was the first attempt to create audience performance developed by our research group. The possibilities of instrument creation with webaudio motivated us to use the audience cellphones as spread loudspeakers taking active part of the soundscape of the performance. Certainly, we had some technical issues to struggle with and we found several solutions learning different lessons, as presented before. We estimate that about 100 people took part of the performance in our first presentation and that is the reason to the reported problems. Now, we want to bring some non technical considerations about the performance.

Due all this issues caused by the number of participants, certainly some people could not connect to the instruments and it resulted in an interesting side effect. At some point of the performance, people did not know if they should try to use their cellphones and take part of the performance, or just watch and enjoy it paying attention in what was happening around. It resulted in an interesting metaphor of contemporary life, where the anxiety of being connected all the time sometimes deprive us from observing what is happening around.

The participation of the audience was an interesting social experience and, somehow, in a unexpected level to us. Some audience members reported that they would like to take pictures of the show but they could not do it since they were using the cellphone to play the DMI. Other person asked for a cellphone charger, afraid to be out of battery before the end of the performance. One person complained that he was texting other friends during the performance and no one answered because their cellphone were been used to play with us. The experience of having the cellphone "stolen" or "kidnapped" for a while had certainly a social impact to some people.

The creation of the performance involved up to 20 people from different areas, knowledge and skills. Also, people with different levels of experience in live performance. Even so, the differences here were added up to make the performance viable. The integration of such a huge interdisciplinary and heterogeneous team was an amazing experience to all the participants. During the performance, the computer guys were struggling with the technical questions, improvising, coding and setting up the server, finding fast solutions to network jam problem, understanding and solving computer problems on the fly. There was always an audience member asking for help, trying to do something while the show was happening. At the same time, the performers and the musicians were also improvising, dancing, playing and acting with the situation, in the middle of the public, keeping it rolling because, at the end, the show must go on and it was live. Maybe only a open live performance with audience participation can give the opportunity to try out the improvisation at this level.

6 Acknowledgments

Authors would like to thanks to all ALICE (Arts Lab in Interfaces, Computers, and Everything Else) members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq and FAPEMIG.

References

[1] Cleomar Rocha. O imaterial e a arte interativa. *DOMINGUES, Diana. VENTURELLI, Suzete.(org.). Criação e poéticas digitais. Caxias do Sul: Edusc*, pages 27–31, 2005.

[2] Benjamin Taylor. A history of the audience as a speaker array. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 481–486, Copenhagen, Denmark, 2017. Aalborg University Copenhagen.

[3] Eran Egozy and Eun Young Lee. *12*: Mobile phone-based audience participation in a chamber music performance. In Thomas Martin Luke Dahl, Douglas Bowman, editor, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 7–12, Blacksburg, Virginia, USA, June 2018. Virginia Tech.

[4] Abram Hindle. SWarmed: Captive portals, mobile devices, and audience participation in multi-user music performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 174–179, Daejeon, Republic of Korea, May 2013. Graduate School of Culture Technology, KAIST.

[5] Sang Won Lee and Jason Freeman. echobo : Audience participation using the mobile music instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 450–455, Daejeon, Republic of Korea, May 2013. Graduate School of Culture Technology, KAIST.

[6] Marcelo Gimenes, Pierre-Emmanuel Largeron, and Eduardo Miranda. Frontiers: Expanding musical imagination with audience participation. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, volume 16 of *2220-4806*, pages 350–354, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[7] Luke Dahl, Jorge Herrera, and Carr Wilkerson. Tweetdreams : Making music with the audience and the world using real-time twitter data. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 272–275, Oslo, Norway, 2011.

[8] Leshao Zhang, Yongmeng Wu, and Mathieu Barthet. A web application for audience participation in live music performance: The open symphony use case. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, volume 16 of *2220-4806*, pages 170–175, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[9] Antonio Deusany de Carvalho Junior, Sang Won Lee, and Georg Essl. Understanding cloud support for the audience participation concert performance of crowd in c[loud]. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, volume 16 of *2220-4806*, pages 176–181, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[10] Bastiaan van Hout, Luca Giacolini, Bart Hengeveld, Mathias Funk, and Joep Frens. Experio: a design for novel audience participation in club settings. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 46–49, London, United Kingdom, 2014. Goldsmiths, University of London.

[11] Luca Turchet and Mathieu Barthet. Demo of interactions between a performer playing a smart mandolin and audience members using musical haptic wearables. In Thomas Martin Luke Dahl, Douglas Bowman, editor, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 82–83, Blacksburg, Virginia, USA, June 2018. Virginia Tech.

[12] S. Astrid Bin, Nick Bryan-Kinns, and Andrew P. McPherson. Skip the pre-concert demo: How technical familiarity and musical style affect audience response. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, volume 16 of *2220-4806*, pages 200–205, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[13] Lewis Carroll. *Alice Adventures in Wonderland & Through the Looking-Glass*. CreateSpace Independent Publishing Platform, 2018.

BUZU: Retrieving public transportation data for sonification

Julián Jaramillo Arango[1]*, **Esteban Viveros Astorga**[1], **Fernando Iazzetta**[1], **Cristiano Figueiró**[1]

[1]Núcleo de Pesquisas em Sonologia – Escola de Comunicações e Artes, Universidade de São Paulo
Lúcio Martins Rodrigues, 443 - Butantã, São Paulo - SP, 05508-020, Brasil

julianjaus@yahoo.com, emviveros@gmail.com, iazzetta@usp.br, figocris@gmail.com

Abstract. *This paper discusses the conception, design and exhibition of BUZU, an audiovisual installation that generates an auditory image of the São Paulo bus transportation system. BUZU makes perceptible information of both the system's planning and behavior during a particular week in October 2017. The work is an artistic outcome of the InterSCity project, an inter-institutional research initiative concerning the Future Internet and the Smart Cities. Along with the discussion of the BUZU creation process we will examine mining and processing strategies related to the sonification of big data, the data-to-sound mapping methods, the auditory structure for displaying the material and the public exhibition of the work in the context of an artistic event.*

1 Introduction

BUZU is an audiovisual installation risen as an artistic outcome of the InterSCity project [1], which designs computational strategies and tools for the Future Internet and the Smart Cities [2]. BUZU was launched in the "Sons de Silício" Art exhibition [3], that was held in the older São Paulo Museum of Contemporary Art in April 2019. BUZU emerged as the result of collaboration between the Research Center on Sonology (a.k.a NuSom) and the InterSCity.

Figure 1: The BUZU installation

The installation proposes an acoustical image of São Paulo by retrieving information of the city public bus transportation system, which comprises 2.183 lines. BUZU makes perceptible a dataset created by the InterSCity project, which reports the system's behaviour in a particular working week in October, 2017. While the dataset original purpose is comparing the system behaviour with the Easter holiday, in BUZU the dataset is implemented to feed an audiovisual engine created in Pure Data (a.k.a. Pd) and Processing. The audio is projected by a quadraphonic system and the visuals are displayed in a central screen. Textual information is presented in four little LCD screens distributed around the central structure. By exploring alternative representations of the city, the project adopts sonification as the main strategy.

It is worth mentioning that sonification has been increasingly implemented in projects creating acoustical representations of the city. These projects involve the displaying of the London subway system information with reference sounds [4], the exploration of mapping techniques used in urban planning and design [5] and the organization of geospatial data in geographic maps [6]. An inspirational project is the metrosynth [7] which creates a sonification system of the Montreal subway system running in HTML5.

Instead, Buzu' deals with a huge volume of data, which brought a challenge for our research group, since past sonification projects faced by the members [8-10] coped with smaller sets of information. In order to facilitate data-to-sound and data-to-process mapping tasks, we implemented our work in Pd. Although it made us to create a new data set adapted from the original one, it also allowed us to distribute the audiovisual processing in four Raspberry Pi (aka RPI) units.

The first section will examine the information contained in the original dataset, the characteristics of the one we created and the specifications of BuzuDados, the Pd abstraction which parse our dataset. The second section will report the data-to-sound mapping strategies implemented in the BUZU engine, by describing the behaviour of time, the drone sound synthesiser and the melody generation machine. The visual display and the multimedia hardware structure used in the first version of BUZU as well as the context in which it was exhibited will be discussed in the third section. The last section will outline some conclusions and future work.

2 Parsing the dataset

2.1 Analysing the original dataset

The original dataset and its documentation are available in the InterSCity website [1], [11]. The "Bus movement model", as it is named, is a 146Mbytes file with 8 pairs of XML files intended to be incorporated in the InterSC-Simulator [12]. It represents trips performed by 2,183 bus lines in São Paulo. The information feeding the dataset

*Supported by CAPES.

came from two sources: GTFS (*General Transit Feed Specification*) and AVL (*Automatic Vehicle Location*). The former reflects the service planning by providing data such as bus line code, route, bus stops location and pathway between stops. These open data can be consulted on the SPtrans public site [13]. The latter reflects the system`s real behavior by providing data such as departing time, departing frequency and average speed. These data are gathered by GPS devices mounted on each vehicle, and were provided by Scipopulis [14], a startup devoted to process São Paulo transportation data from the Olho Vivo system [15].

Seven of the eight pairs of XML files correspond to a single day of a “typical” week (22 to 28 October, 2107), that is, a week in which the system operated without interruptions. The extra pair corresponds to an atypical day, such as the Easter holiday, when the system presents anomalous behavior.

The first file of each pair, called *buses.xml,* comprises GTFS data, while the second one is called *maps.xml* and comprises information from both sources. The two XML files interact each other, since in the *buses.xml* the bus stop locations are coded as a ten digits long number called node, while in the *maps.xml* every node corresponds to a couple of Lat and Long coordinates. It means that the software exploring the dataset should be able to shift between the two files in order to find data related to a single bus line on a single day.

2.2 Creating the NuSom dataset.

For BUZU we adopted Pd as the main software tool, since it can be run on the RPI. It would keep the project`s costs lower and would allow us to bring the Sao Paulo transportation system`s data to an open sound-making arena. With this in mind, our next step was trying to parse in Pd information from the InterSCity dataset.

We chose bus stop locations as the first data to be retrieved because they can be checked on online platforms such as Open Street Maps [16]. This empirical method were helpful to verify whether large amounts of parsed data were consistent with other maps.

Our first try was with the POF external library [17], which has an xml-parsing object. However, we failed retrieving the data since the XML files seemed to be too large and their arriving time was rather erratic. Another problem we faced was data loss. Pd (0.49-0) have a single precision floating point of 32 bits, which truncates large numbers such as those concerning with bus stop locations. In view of these difficulties we opted for creating our own dataset, to be used in Pd with the help of other programming tools. The limitations with single precision of Pd have been discussed [18] in the Pd developer community. The solution for a double precision resolution has not yet been adopted by the issue of code compatibility between earlier versions and externals produced by the community. Our solution here was adopt other programming techniques to adapt the dataset to fit with Pd capabilities.

Inside Pd’s parsing possibilities we started trying to use the Cyclone library *coll* object [19] to host data from the xml documents. When the tests started to fill the object buffer with large amounts of data we began to identify some data loss so we start to use the *text* object which suits better for the task and keeps the compatibility with Pd vanilla.

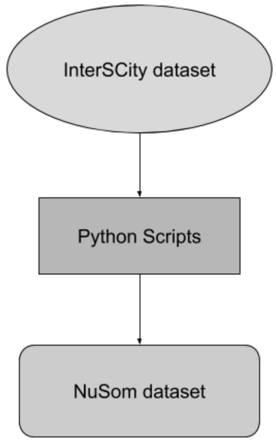


Figure 2: Dataset diagram

With Python 3 and the Pandas library we created three scripts that extracts some information from the originals *buses.xml* and *maps.xml* as well as from other intermediate files created by us, and distribute these data in a new set of files. Our goal was that the new dataset, named NuSom dataset could be readable by the *text* object.

The first python script, *cria_onibus_dia_pd-vanilla.py* generates 1 file per day. It parses the 8 original *buses.xml* and creates 8 new files. The former was created to adapt the new dataset to the *text* object operation together with *list split* object. The dataset display GTFS data such as departure interval, start time and bus stop locations.
These last data were successfully retrieved appending *id* characters to the original longer integers. This bus stops identifier prevented from data loss and resolved the precision floating-point issue.

The second script, cria*_trajetos_dia_pd-vanilla.py* generates 1 file per day. It retrieves data from the *map.xml* files and creates 8 files. They are named *trajetos* and display the total distance in the course in meters and average speed at a set time once a trajectory identifier has been provided. Most of these identifiers correspond to bus lines trajectories, which consists of the first and last bus stop separated by a dash.

The third script *cria_coordenadas_dia_pd-vanilla.py*, works in a different way. The script generates 8 files named *coordenadas* by retrieving information from some files created for us in an earlier phase of the project. These files are called *map_id_x_y-latlong*, and were obtained manually from the *map.xml*. We got the Lat and Long coordinates by converting the original nodes coded in UTM (Universal Transversal de Mercator) using an online converter [20] and a flexible text editor. Then each Lat and Long coordinate was scaled from 0 to 1000 in order to enable the creation of a meaningful file for sonification. By associating the correct bus stop identifier (the large number + the *id* character), the script *cria_coordenadas_dia.py* creates pair of scaled coordinates for each bus stop. It resolved the single precision floating-point issue. Furthermore, the script should be able to retrieve city region data from the postal code system, but this feature is under implementation.

2.3 BuzuDados specs

To navigate in our dataset we patched the buzuDados abstraction. It has two inlets and two outlets and works by demanding information with messages sent to the left inlet. Data is retrieved in the left outlet. buzuDados starts working when sending a number from 0 to 7 in the right inlet corresponding to the day of the week or the Eastern holiday. The right outlet will bang after data is retrieved. The left inlet and outlet works as follows.

When sending a number from 0 to 2.183 corresponding to the *lista_onibus* file, buzuDados retrieves the string <bus id>, which corresponds to the unique bus line identifier according to the SP trans system. When sending a message with <bus id> and then <start_time>, buzuDados returns a string symbol corresponding to the time in which the bus line should start operating; <bus id> and the string <interval> a list of 24 numbers, each one corresponding to the departure frequency (in seconds) for each of the 24 hours of the day; <bus id> and the string <stops> a list of strings corresponding to the identifiers of each bus line stop.

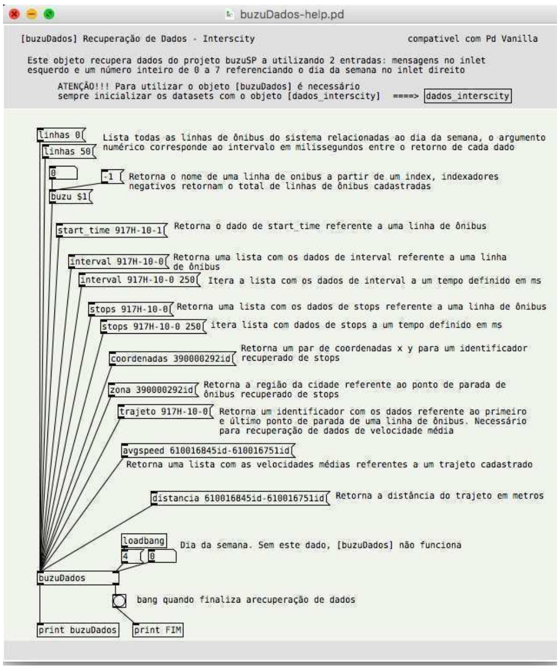


Figure 3: The buzuDados help.pd

When sending a message the string <coordinates> and a bus stop identifier, buzuDados retrieves a list of 2 numbers <x> and <y> corresponding to the scaled Lat and Long coordinates of this bus stop. When sending <zona> and then a bus stop identifier, it retrieves a number from 0 to 4 corresponding to São Paulo zones: Downtown, Western, Eastern, Northern or Southern.

Lastly, when sending a message with the string <avgspeed> followed by the first and last bus stop identifiers of a line separate by a dash, buzuDados retrieves a list of 24 numbers corresponding of the average speed (in m/s) at each of the 24 hours of this day The NuSom dataset and the buzuDados abstraction are open and can be downloaded in our GitHub repository [21].

3. Data-to-sound mapping

Simultaneously with the process of constructing and refining the dataset retrieval and analysis, we conducted a sort of experiment in the design of the sonic content. The goal here was finding the sound poetics suitable for exploring the passage of time within the public bus transportation system, which could be reflected in a listening environment. Although BUZU did not implement any interaction device, it does provide a synesthesia experience between the projection of the bus lines highlighted on the map and the sound being generated. The sound mapping strategies were defined taking into account the possible struggles that people could experience during a bus trip. This would be experienced by contemplating the data visualization and sonification present in the installation.

The BUZU audio engine is composed by two synthesisers working simultaneously: the drone and the melody maker machine. The former operates a an acoustic background, a lower-spectrum pad sound giving the sensation of continuity. It retrieves AVL data. The latter sonifies the route of up to four randomly chosen bus lines, by tracking the path followed at each bus stop. It produces identifiable melodies and retrieves mainly GTFS data. The drone and the melody maker machine receive data from buzuDados abstraction and are driven by the transport device.

3.1 Time

Our strategy to manage the passage of time in BUZU was using a Transport device, which is a time management interface widely used in DAW (Digital Audio Workstation) devices, audio recording and editing software. It is in the transport that can be found the play, pause and stop buttons. Once activated, a count is started by the Transport device with precision of 100 milliseconds. Through successive divisions, a count is generated where we have output such as day, hour, minute, second and millisecond.

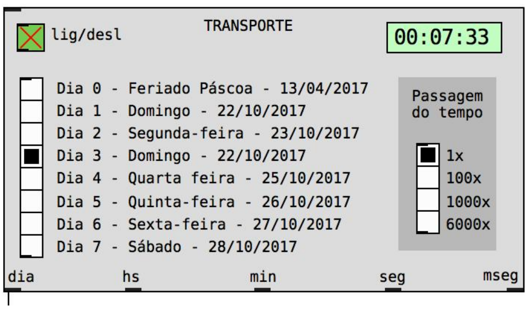


Figure 4: BUZU transport device

It is possible to change the day and the speed. These features facilitate navigating the dataset over time, making possible to select the day and time when the dataset is queried.

3.2 The drone synthesiser

The drone synthesizer sonifies <distance> and <avgspeed> data, referring to the bus line pathway contained in the NuSom dataset. The strategy adopted was to use three overlapping textures to generate a predominantly contemplative sound structure.

The first texture is a sort of subwoofer, we used this layer in order to establish a synthetic and timeless acoustic space for the installation. We employ the *lfnoise~* generator and *else* library filters in order to get a subtle and deep texture. Since the audio system is quadraphonic, we chose the low frequency noise generator with fixed seed in order to produce monophony between the two pairs of stereo audio systems. The existence of this layer, is independent of the dataset, since it fulfills a role of aesthetic establishment of the acoustic space that the work intends to install.

The second texture is similar to the Risset cascade of arpeggios [22]. By implementing additive synthesis, we generate a sonic spectrum that, through small alterations of pitch, is able to modulate its harmonics and promote sound beats.

In this texture we intend to produce states of restlessness and relaxation by manipulating the internal beat rate of the sound spectrum according to the average speed of the bus line at a given time of the day. This rate is obtained by calculating the weighted harmonic average [23] of the <avgspeed> in the complete route (from the first to the last bus stop) of a given bus line. The weighted harmonic mean velocity is calculated by making use of the average speed data of each path at a given time and the distance of each bus line path. The ratio between the average bus line speed in the time of day and the weighted harmonic mean of the line speed in the full path generates the number representing the average speed variation rate which, after being scaled, will control the density/velocity of the beats between harmonics in the second texture.

In this way, the slower the movement of the specific bus line, the greater the interference of sound beats in the generated tone. This mapping seeks to mimic in the sound plane a slow and possibly uncomfortable passage of time inside a bus in a busy traffic. The movement of the harmonics in this model generates very noticeable results, and potentially communicate the density of traffic in the sound domain.

Lastly, there is a noise texture whose morphology is dependent on the average speed of each bus line and can only be heard when invalid data is retrieved. In the InterSCity dataset, the <avgspeed> data with a value of -1 indicates that some data sampling failure occurred (failure to transmit, receive or even fail to trigger the GPS), in the NuSom dataset this value. Thus, by having an average speed in the path of -1, the noisy texture will be activated. The average speed of the bus line in the full path feeds the density parameter of the *dust* generator present in the *else* library [24]. It will generate a grainy random texture and can refer to an idea of analog noise, directly signaling a failure in the original dataset.

3.3 The Melody Maker Machine

The melody maker machine is a four-voice polyphonic FM synthesiser attached to a resonant filter using the *bob~* Pd object, and then attached to a spectral delay [25]. The synth is driven by a dynamic ADSR, which receive pulses coming from the Transport device. It has four parameters (pitch, cutoff frequency, resonance, and delay feedback) and four outputs envisaging the quadraphonic arrange.

The synth retrieves information such as <coordinates>, corresponding to the GPS location data of the bus stops achieved by each line in <x> and <y> axes, and <interval>, corresponding to the number of departures per line at each hour of the day. The former was mapped to spatial data by producing an acoustical matrix of 1000 x 1000 values, the latter to the delay feedback parameter.

While our goal was creating an acoustical image of the city, our task was emphasize a sense of direction to each cardinal point rather than an exact location, by making evident direction changes. Thus, we assume a divergent mapping or many-to-one technique, where "...objects usually change their sound characteristics in several aspects at the same time when varying". [26]. In this regard, the <x> and <y> coordinates of each visited bus stop were mapped in two different ways.

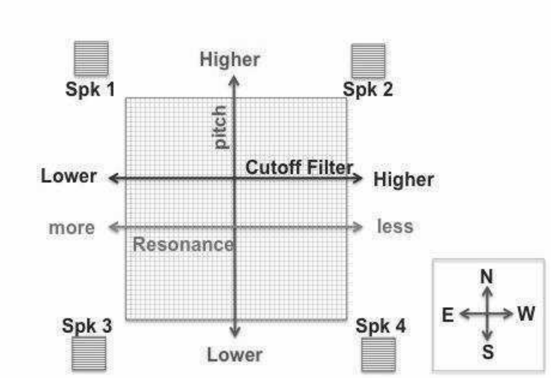


Figure 5: Mapping the <x> and <y> coordinates in the melody maker

On the one hand, the <x> coordinate was scaled and connected to the Cutoff frequency and, in reverse order, to the Resonance parameter. The <y> coordinate, connected to the pitch value, was scaled and redirected in order to avoid chromatic relations by selecting just notes of the Cm pentatonic scale. On the other, the coordinates where connected to the audio output levels, taking advantage of the quadraphonic system to recreate the cardinal directions in the installation space. The east-west axe was associated to the <x> coordinate, and the north-south axe to the <y> one.

4. The BUZU Auditory Display

4.1 Complementary visualization

In BUZU a screen is in the center of the quadraphonic space showing the spatial displacement of the bus lines on the map representing the metropolitan region. The map is rendered with the Processing language connected via OSC to the server buzuDados (running in Pd).

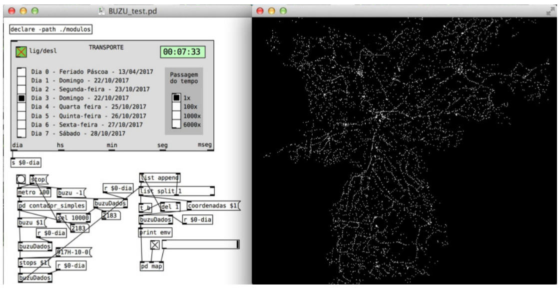


Figure 6: Processing visualization receiving data from the server

Around the central screen there are 4 LCD display connected via the wi-fi Wemos D1 mini microcontroller. The displays showed a text corresponding to the number ID of each bus line highlighted in the map. The main purpose with this visual clues was to provide a reference that triggered the recognition of the acoustic parameters, and then the emergence of a sounding image of the city.

4.2 Distribution of multimedia tasks

BUZU distributes audio synthesis, image generation, and data analysis tasks to four RPI 3 B + units over a Local Area Network (LAN).

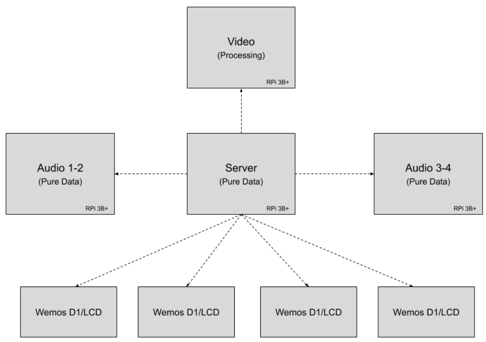


Figure 7: Buzu operational diagram

A RPi play the role of a server, in which the Transport is running, the NuSom dataset is parsed, and the data for sound and image synthesis is generated. The data is sent over the WiFi LAN, using the UDP-OSC implemented in the the Pd *netsend* object. The data is sent through the router's broadcast address, to the other 3 RPI and the 4 WiFi Wemos D1 mini microprocessors. There are, in sum, seven clients.

The Audio 1-2 and Audio 3-4 RPI units are responsible for quadraphonics, with Audio 1-2 being responsible for the front stereo pair and

Audio 3-4 for the stereo rear pair. The Video RPi is responsible for the screen real-time rendering of the bus stop data used in the sonification and finally the Wemos D1 mini are responsible for receiving and displaying the ID of each retrieved bus line

5. Conclusions and future work

By taking advantage of sonification techniques BUZU offers a poetic experience in the perception of the passage of time by contemplating urban traffic. While visiting the installation, it is possible to experiment a kind of poetic enchantment by the contemplating the complexity of the system. At the same time, an image of the city start emerging when the visitor realize the relation between sound and the dynamic map. In this regard, the project goal was completed, since our intention was to generate an alternative view of Sao Paulo by retrieving data from the bus transportation system

However, although the installation worked very well in its public exhibition, and, the performance of the buzuDados abstraction retrieving data from NuSom dataset run well, we can envisage some adjustments. One of them is concerned with the feature capable of create interruptions in the calls made to the dataset. In addition we plan to implement internal messages in the control of the buzuDados dataflow. It will enable the expansion and maintenance of data parsing features in a more robust way, facilitating the abstraction implementation by third parties.

Although buzuDados completes specific operations for BUZU, it deals with processes relevant for other sound designers concerned with big data sonification. buzuDados makes feasible for others the implementation of the InterSCity dataset in Pd. It is expected to create alternative sonifications of the NuSom dataset using the buzuDados abstraction in collaboration with other members of the research group. Furthermore, a closer collaboration with InterSCity members is also planned, regarding the implementations of other datasets.

Acknowledgments

This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, FAPESP proc. 14/50937-1, and FAPESP proc. 15/24485-9.

References

[1] InterSCity. InterSCity Website. http://interscity.org/

[2] Macêdo Batista, D; Goldman, A; Hirata, R Jr.; Kon, F; Costa, F and Endler, M. (2016) InterSCity: Addressing Future Internet Research Challenges for Smart Cities. 7th IEEE International Conference on Network of the Future.

[3] Sons de Silício. (2019). Art Exhibition Website. https://sonsdesilicio.wordpress.com/

[4] Nickerson, L.V., Stockman, T., & Thiebaut, J. (2007). Sonifying the London Underground Real-Time-Disruption Map. Proceedings of the 13th International Conference on Auditory Display, Montréal, Canada, June 26 - 29, 2007

[5] Adhitya, S & Kuuskankare, M. (2011). The Sonified Urban Masterplan (SUM) Tool: Sonification for Urban Planning and Design. Proceedings of the17th International Conference on Auditory Display (ICAD-2011) June 20-24, 2011

[6] Brittell, M. (2018). Seeking a reference frame for cartographic sonification.Proceedings of the 24th International Conference on Auditory Display June 10 -15, 2018

[7] Gover, M. (2018) https://www.matangover.com/metrosynth/

[8] Arango,J,J. (2018) AirQ Sonification as a context for mutual contribution between Science and Music. Música Hodie. V.18.N.1. pp: 92-102. DOI:10.5216/mh.v18i1.53573

[9] Arango,J,J. (2018) Esmog Data: Interpreting Air Quality Through Media Art and Design. Leonardo Electronic Journal ISSN: 1559-0429 ed: v.51 fasc.N/A p.183 - 183 DOI: https://doi.org/10.1162/LEON_a_01515

[10] Iazzetta, F; Piccinini, A. (2018) Sonificação como art-science: três lógicas de interdisciplinaridade, três lógicas de sonificação. Anais do XXVIII Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música. Manaus.

[11] Wen, M. etal.(2018) Criação de Modelo para Simulação de Movimentação de Ônibus a Partir de Dados Reais. I Workshop Brasileiro de Cidades Inteligentes.

[12] Santana, E. F. Z.; Lago, N; Kon,F and Milojicic, D. S. (2017) "Interscsimulator: Large-scale traffic simulation in smart cities using

erlang," in The 18th Workshop on Multi- agent-based Simulation

[13] SP Trans. Website http://www.sptrans.com.br/desenvolvedores/GTFS.aspx\

[14] Scipopulis Website. https://www.scipopulis.com/

[15] Olhovivo Website. http://olhovivo.sptrans.com.br/

[16] Openstreet Maps Website. https://www.openstreetmap.org

[17] POF - Pd + Openframeworks. https://github.com/Ant1r/ofxPof

[18] Katjaa, V. Double precision in Pd. 2011. http://www.katjaas.nl/doubleprecision/doubleprecision.html

[19] Cyclone library. Repository. https://github.com/porres/pd-cyclone

[20] Leware Website. UTM to LatLong converter. https://leware.net/geo/utmgoogle.htm

[21] Viveros, E. Arango, J. J. & Iazzetta, F. NuSom dataset. https://github.com/LABNuSom/Interscity_pd_dataset

[22] Risset, J. C. (2003). Computer Music: Why? https://liberalarts.utexas.edu/france-ut/_files/pdf/resources/risset_2.pdf. Retrived 2019-05-26. Online pdf document.

[23] Wikipedia. Harmonic mean. Acessado em: 28/05/2019. https://en.wikipedia.org/wiki/Harmonic_mean#Weighted_harmonic_mean

[24] Porres, Alexandre Torres. Else Library for Pure Data. 16th Brazilian Symposium on Computer Music. 2017. http://compmus.ime.usp.br/sbcm/2017/papers/sbcm-2017-6.pdf

[25] Barnecht, F. (2012) Guitar Spectral Delay. https://guitarextended.wordpress.com/2012/02/07/spectral-delay-effect-for-guitar-with-pure-data/

[26] Grond, F&, Berger, J. (2011). Parameter Mapping Sonification. In: Herman, T., Hunt, A & Neuhoff, J. G. The Sonification Handbook.

Composing through Interaction: a framework for collaborative music composition based on human interaction on public spaces

Mauro Amazonas[1] , **Thais Castro**[1] , **Rosiane de Freitas**[1] , **Bruno Gadelha**[1]

[1]Universidade Federal do Amazonas
Av. General Rodrigo Octávio, 6200 – 69080-900 Manaus, AM

[mauro.jr,thais,rosiane,bruno]@icomp.ufam.edu.br

Abstract. *Urban public art is a kind of art that is produced and demonstrated in public places, based on the function and connotation of the city itself exerts. As an essential artistic content in the contact of human life, the introduction of technology is a significant trend in public art, and with it, the interaction has become an increasingly relevant aspect of public art in the digital context. In this way, this work presents an environment for creating random collaborative music from interaction in public spaces using mobile technology. The result is a composition that goes towards to John Cage's methods. However, in our case, all participants are composers and their interactions with space work as the component that brings randomness to composition. A case study was conducted with volunteer students divided into groups. Participants made use of two versions of Compomus - an app developed for immersive interaction with sound. One version encourages movement through the environment, while the other explores the spatiality of sound in a simulated public environment within the university. The interaction of the participants generated ten compositions, five from the first version and five compositions from the second version of the developed application. The sounds resulting from the interaction were made available to the public through a website.*

1 Introduction

Public spaces refer to places where people can get along and interact in society. Cattell *et al.* [1] say that these spaces are a fundamental feature of cities and their quality is often perceived to be a measure of the quality of urban life. These spaces offer some facilities and services that Jiang *et al.* [2] divide into four categories as follows: (1) health safety facilities, such as public restrooms, street lamps, etc; (2) leisure facilities, such as newspaper stands and kiosks; (3) information communication facilities such as traffic signs, bus stations, etc; (4) art services facilities such as flowerbeds, landscape, etc. Urban public facilities, as an integral part of the city, reflect urban development and urban culture, on the other hand, connect with the urban environment and the public to form a "human-environment" system. When digital media technology is introduced, public facilities and citizens form a system of the interactive experience of "user behavior-environment-technology" [2].

In this context, the experience of users in public spaces can be enriched when we mix leisure and arts facilities with technology to create an interactive environment for people. To foster public interaction in these spaces, public art in a digital context emerges as an alternative given that this type of art emphasizes participant subjectivity, participation, bidirectionally, and feedback. Digital public artworks differently from traditional public art, that is, the artist does not entirely control the content, and the power of creativity is in the hands of the public. In this process, interaction requires that artists give up creative power and objects to be enjoyed freely by the public [2].

Music and Sounds are forms of artistic expression that is often realized in public spaces. Sound Art or Sonic Art (SA) encompasses very diverse types of creation based on the exploration of sound ("literal or implicit") as the main element, whether in combination with more traditional arts such as painting and sculpture or with digital and interactive media. It is a mutant definition, as it includes more and more emerging genres, in addition to sculptures and sound installations, electroacoustic music, acoustic, algorithmic, computer music, and noise [3].

Mobile devices are available computational devices that are already disseminated and can be explored to provide human-human, human-environment, and human-environment-human interaction [4]. They bring mobility, processing power, connectivity, and relatively low cost. When analyzing their multiple functionalities, it is realized that these devices can be considered of general-purpose and have the capacity to mitigate costs for simple solutions like controlling reproduction of a player or even more complex like a collaborative music composition through an Application like Trackd [5].

This paper presents research that aims to explore SA as a public art installation. In this installation, people can interact with the environment through sounds using mobile devices resulting in a collaborative electroacoustic music composition approach. The main characteristic of this approach is its unpredictability that is achieved by the free interaction of the users in the public space. This issue gives us the random component of the strategy and the resulting music goes towards John Cage's random style [6], but unlike him, we don't have a single, but multiple composers. This fits the concept of public art in the sense that we don't have control over the user's interaction in the environment, and the resulting collaborative music is always unique and unpredictable. The resulting composition is a piece of electroacoustic music that, according to

[7], this kind of music is composed of previously recorded sounds, which are later treated in a computer and worked musically to build the final product (work).

The remainder of the paper is organized as follows. Section 2 presents some concepts about the field of study of sonic interaction design and its application in this work. Section 3 presents examples of applications that make use of technology to enable collaborative music creation. Section 4 explains some concepts and possibilities in the exploration of the spatiality of sound for the artistic field. Section 5 presents the materials and methods used in the development of this research. Section 6 presents the Compomus app and its versions used during this study. Section 7 explains the case study carried out to evaluate the acceptance of the proposed technology. Finally, the conclusions of the study are presented in Section 8.

2 Sonic Interaction Design

The way to interact and how to develop the sound design of products or artifacts are studied by Sonic Interaction Design (SID), an emerging field of study that is situated at the intersection of auditory display, interaction design, ubiquitous computing, and interactive arts. SID can be used to describe practice and research in any of the various functions that sound may play in the cycle of interaction between users, artifacts, services, or environments [8].

A classic example of the SID application was the first iPod model, which used a mechanical scroll wheel as the input device: a rotating wheel to allow scrolling between menu items. The element that replaced the mechanical wheel used on iPods is the clicker: a click sound that provides feedback for movement between menu items. This feature gives a tactile feel to the click wheel (a pseudo-haptic illusion), somewhat similar to the turntable on old phones, making scrolling more expressive and more informative. The click sound is the only sound produced by the iPod outside of the headphones and is generated through a small piezoelectric speaker inside the device [9].

New forms of interaction with sound have been presented using technologies, playful, artistic, and creative experiences that are provided through the relationship of art to science and technology. Based heavily on SID concepts, the authors in [10] developed interactive interfaces that use ReactVision, software to view tags and control sound. From this technology, several concepts of the study have experimented in different scenarios with individual musical performances, group and collaborative applied in schools and art exhibitions.

In this work the SID is applied as follows: the user assumes one of the sounds available in the application used as his identity in the interaction space, what happens is the "sonification" of the participant. This personification of the sound occurs in both of the app solution purposed. The interaction of users with the environment using one of this versions takes place through the movement, which is captured and processed by the smartphone sensors and sent to the audio server. The now "sonified" user has their sound played in the speakers and can interact with other users while inside the interaction space. In the other version, the user has control of the reproduction of their sound as well as their spatial location to interact with other sounds and the environment.

3 Creating Collaborative Music with Emerging Technologies

The human being is very creative and always looks for different ways of making music, whether mixing rhythms, languages or even using different types of instruments. Music composition is a naturally collaborative process, and new technologies and internet infrastructure improvements enable this collaboration in music production and composition to be performed by countless people in different physical locations.

However, while technological advances allow such exploits, technology seems to be individualizing the human being even in activities that were previously practiced together. An example would be the production of a song, in which it was only a short time ago that the artists traveled to another country to produce their album. Today this is a thing of the past, using software like the Pro Tools Collaboration [11], artists from different parts of the globe can work on a project together without leaving home. While breaking borders and allowing interaction with people in multiple places, technology has also kept us from personal relationships in the same environment, today we can do a lot without leaving home and interact less and less with each other in "old" style [12].

Against the current technological trends, our solution focuses on the process, in interaction and collaboration of the participants in the same place spontaneously and encouraging communication and organization by the participants themselves. The environment allows users to be autonomous and independent in creating music using their smartphones. The technology, in this case, functions as a means for participants to more easily interact and observe the impact of their actions on the environment.

4 Real-Time Sound Spatialization

The human being can determine the location of a given sound, due to the hunting ability developed by his ancestors. This ability works by employing auditory indications that are perceived by the auditory system. According to Lord Rayleigh [13], the interaural time difference (DTI) and the interaural intensity difference (DII) are the most commonly used indications. The DTI indicates the difference in time that the sound wave takes to reach each ear and the DII the difference in intensity. It is understood that the auditory system, based on these indications and more specific ones (such as the order of arrival of the waves and the spectrum), considering a complex sound, determine the position (direction and distance) of the source through a voting system [14].

In acoustic music performance, there are no musicians positioned on a stage, and the reproduction is performed in an arrangement of loudspeakers positioned in the presentation place and around the listeners. In most presentations, it is the composer who controls a sound mixer placed in the center of the presentation location, processing and directing the sound in real-time to speakers and thus shaping the space [15]. In this way, there is a mobility of sound around the audience, creating an exciting surrounding effect. The audience usually feels that they are immersed in sounds. The artist who controls the direction and mobility of sounds in space through loudspeakers creates the sound spatialization effect (sound diffusion or electroacoustic diffusion).

5 Methodology

For this project, a literature review was carried out to know the state of the art, tools, and technologies that could be used. In this search, we have observed several tools like OSC protocol, Pure Data, and approaches such as sound spatialization and Sonic Interaction Design (SID). Based on these observations, Pure Data, SID principles and sound spatialization were incorporated into the project, thus expanding the theoretical, technological and artistic possibilities of this work. In the Pure Data platform was developed a framework for the reproduction and sound spatialization on the Open AUDIENCE architecture for an immersive auralization to provide a differentiated experience for the user [16]. The Ambisonics auralization technique was chosen for use in this work because of its easy implementation, low computational cost, flexible speaker configuration, capable of supporting multiple people simultaneously and good directionality of virtual sound sources. For the mobile side, the mobile development platform chosen was Android as the most used platform in the world [17] and for its ease and freedom of development.

To achieve the intended results in this study, two versions (with different goals) of the application called Compomus for Android were implemented. The first, PlayStop version, detects the user's presence in space and plays their chosen sound without any further intervention. Playback terminates when the App detects that the user has left the space set. In the second version, JoyStick allows the user beyond the control of sound reproduction, control the spatial location of the same. To observe the environment usage, a case study was carried out with the volunteer students using the two versions created in a space defined within the university. This study generated five musical compositions and their results can be checked in [18].

The material used to perform a musical "performance" are devices easily found in our everyday life with low costs such as a standard internet router, a generic USB sound card with 5.1 channels, cables, four portable loudspeakers, and a notebook. These equipment are the requirements for a quadraphonic musical composition, in which it is possible to explore sound spatiality in this case. There is also the possibility of making use of only two speakers in stereo mode. Further details of each of these components can be observed in the following sections.

6 Compomus

This research originated from the demand for a partnership between the Faculty of Arts and the Institute of the Computing of the Federal University of Amazonas. The idea of this partnership was to join the Arts (especially music) with Technology, addressing the problem of the composition of music with the use of emerging technologies.

The scope of this work is in electroacoustic music (more precisely in live electronics). The interaction of users and their collaboration through their smartphones become an integral part of the work as a source of originality for the composition of musical works intended to be performed in real-time in public spaces. With the participation of an undetermined number of people, the composition happens through the contribution of each person with its sound. Any participants can organize the composition or not, also allowing their intercommunication. All this interaction with the system is processed, reproduced, and recorded in real-time.

The main idea of the proposed system is to allow users to cooperate in composing a sound through their free interaction with other users, and with the sound, they choose to represent them. The composition space can be a room or an open public space with a limited area. Sound speakers are needed to reproduce the sounds of the users who are within this space. The dynamics of system use is as follows: once a user is within the defined area, their sound is played in a loop, if the volunteer moves away from the assigned space, the system interrupts the playback of your sound. What defines whether or not the sound will be reproduced is the user's distance from the center of the interaction space. The speakers play a critical role in the system as they are responsible for audio feedback to the users. Users entering and leaving the interaction space turn the sound on and off on the speakers.

As previously mentioned, the App Compomus was developed on the Android platform and functions as a controller for an audio server that processes all the commands sent by the App executing the sounds. This audio server was developed in Pure Data and uses the Open Audience Architecture as a sound spatialization engine and receives commands through the network. A web server was also designed to control and register users. To support the scenario described above, we have developed a diagram that demonstrates the dynamics of the environment represented in Figure 1, which comprises: A server with a dual function, web server, and audio server. Four amplified speakers connected to the audio server for feedback in the interaction space. A router, which functions as an access point and allows the connection between the application and server. In the PlayStop version, the radius set is calculated by the App that can determine whether or not the user is in the interaction space. When the application detects that it is within the interaction space, the sound chosen by a user is played in the speakers.

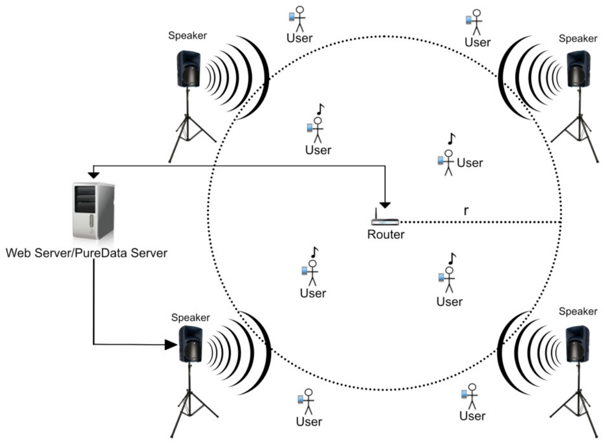


Figure 1: System Function Diagram.

When the version used is the JoyStick, the defined radius is scorned and from anywhere within reach of the wireless network, it is possible to reproduce the chosen sound besides also be able to direct it to the soundbox of your choice. More details of the two versions are presented in the following subsections.

6.1 Compomus PlayStop

The App Compomus PlayStop is the implementation of the idea initially proposed in the project. However, during the researches, the possibility of exploring the spatial sound also appeared, being necessary to separate the two solutions so that these were evaluated separately. The PlayStop version works pretty simply, in Figure 2 it is possible to check the screens of use of the App where it is first necessary to make a user registration and choose a sound. Just as in the JoyStick version, there are 50 sounds of the electronic genre available for free on the LANDR website [19].

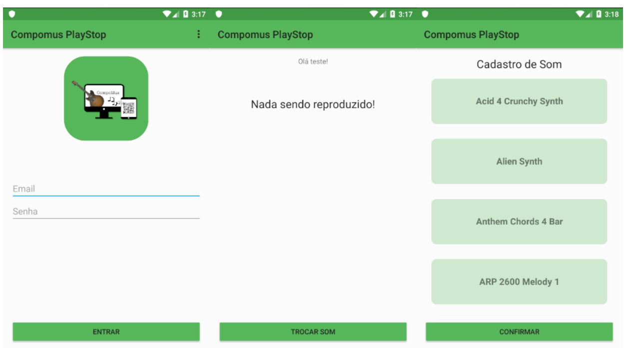


Figure 2: Main screens of the App in the PlayStop version, available only in Brazilian Portuguese.

The second is the main screen of the App in which the feedbacks are presented to the user. The "Nothing playing" message is displayed when the user is not inside the interaction space and the "Playing your sound" is displayed when the app detects that the user is within the defined area. At any moment, the participant can change its sound. For this, there is a button on the main screen that allows such action. The third screen has the list with the available sounds.

In this version, the participant of the composition needs to move, leave the space defined to stop the reproduction, which is intended to stimulate the movement between users. There exist another way to stop the sound. The already known is the environment exit and the other is by clicking to change the sound, the reproduction is stopped but is resumed when the new sound is chosen.

6.2 Compomus Joystick

The Compomus JoyStick is the version of the App that explores sound localization as an artistic element. In which case the user will have control to move the sound in an environment with scattered speakers. The JoyStick version has differences of functionality concerning the PlayStop version since it does not require the user to move through the space Figure 3.

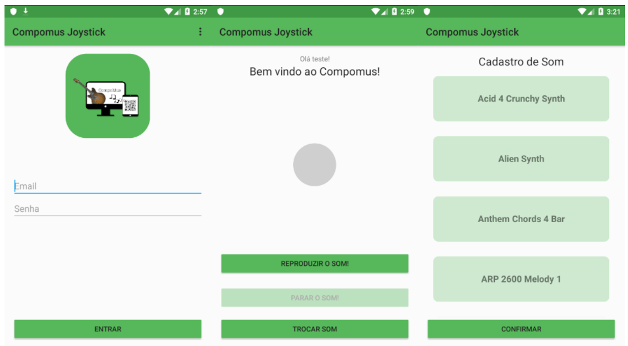


Figure 3: Main screens of the App in the JoyStick version, available only in Brazilian Portuguese.

The dynamics of using the App in this version is very similar to the PlayStop version, the initial screen and register works in the same way as the previous version, as well as in the selection screen of a new sound. The difference is in the main screen (central screen), where a button is available to play and one to stop the sound reproduction and a directional joystick that controls the movement of the sound. There is also the sound switching button that allows the user to change the sound at any time.

7 Case study

A case study was carried out to study musical composition in a scenario in which there is no "central actor", that is, without any particular organization in which one depends exclusively on the collaboration and interaction of the people present in a given environment. This study was carried out in an experimental setting, in an environment simulating an installation in a public space. This methodology was used based on Yin's claim [20] that the case study is the most appropriate methodology to use in research where the main questions are "how?" Or "why?". The author also refers to the case study as an indicated methodology for social studies. Therefore one of the topics addressed in this work.

Also according to the literature [20], as steps for conducting case studies, there are: the **Plan**, which consists in deciding the methodology that will be used for the research; o **Design** or **Project**, where the units of analysis should be defined and the probable case studies; a **Preparation** consisting of one or more pilot case studies; the collection, where the data generated by the pilot study are extracted, and finally the Analysis stage, which consists of an analysis of the collected data. If collection is not sufficient, one may return to the preparation stage for further pilot case studies, or even if the generated data is not desirable, it is necessary to return to the design stage.

Plan: the context that this work is inserted is that of the collaborative sound composition, placing the user as a composer and interpreter of a sound to be built by the individuals themselves collaboratively even if they do not have previous knowledge about musical composition when doing use of the proposed applications.

Design: the aim is to identify improvement points and evaluate users' acceptance of the proposed technology. If the users felt composing a sound even without harmony or synchronism, it was observed that they were collaborating for a musical composition as a whole.

The research involves data collection in the interaction space created within the University. The technological platform developed is new and, as there were no references to assess its viability, it was decided to conduct two pilot case studies first. Data collection was performed by observing the interaction, post-test questionnaires printed and answered by all pilot study participants after each session. For the analysis of the collected data, each pilot case study had some of its most relevant characteristics observed, and its data were collected and discussed in detail in the next section.

To evaluate the results obtained in the analysis and to find out if the technology would be well accepted by the users, the Technology Acceptance Model (TAM) model was used. The TAM model was designed to understand the causal relationship between external variables of user acceptance and actual use of the computer, seeking to understand the behavior of these users through the knowledge and ease of use perceived by them [21]. According to the guidelines of the TAM model, five hypotheses were elaborated that can be checked next. These hypotheses are validated through a questionnaire applied after the use of the technology.

- H1 - Perceived usefulness is affected when there is no collaboration between participants.
- H2 - Intention of Use is affected by cultural influences.
- H3 - Perceived Usability Facilitates engagement with technology.
- H4 - Perceived Usability and Ease of Use provide a positive experience with technology even without knowledge of musical concepts.
- H5 - The ability with technological devices and their resources improves personal performance in the use of technology.

A questionnaire with 30 questions using the Likert [22] scale was developed to validate the questions the hypotheses elaborated. The Likert range has five levels that are defined to capture the points of choice of the users interviewed. In this case, the points of choice are:

1. Strongly disagree
2. Disagree
3. Neutral
4. Agree
5. Strongly agree

The study was conducted within the university premises with the participation of 47 volunteer students who used both versions of App Compomus Joystick and PlayStop. These studies are best described in the following subsections. The two studies were conducted on the same day, and students were divided into groups for physical space issues.

7.1 PlayStop Pilot Study

- **Preparation:** A space of approximately 36 square meters was reserved for our studies as shown in Figure 1. Four portable speakers were plugged through cables to the USB sound card that was used to reproduce the processed sounds in the notebook. This equipment was used as an audio and web server. We also used a standard router, located right in the center of the defined space, which served as the access point used in the study. As this version does not explore the spatiality of the sound, in this study all the sounds were reproduced equally by the speakers.
- **Execution:** participants were first asked to download the App from the app store. Then they were asked to connect the network made available by the system router to register and use the space. It was explained to the participants the operation of the App as well as the dynamics of the process of the interaction space Figure 4.

Figure 4: Participants during the study making use of the APP defining composition strategies among themselves.

Participants were asked to feel free to interact as they wished. User interactions have been recorded in audio and video. However, only images can be checked on the page created for the demonstration of results in photos and audio [18]. Users could use the app in the environment for five minutes in each round of the study; at that time, they were free to interact or not.

7.2 JoyStick Pilot Study

- **Preparation:** to perform this study after the use of the PlayStop version the participants were invited to use the JoyStick version of the Compomus in the same space and to use the same structure of the previous study Figure 5.
- **Execution:** participants were asked to download the App in the JoyStick version. In this case, it was no longer necessary for a new registration since both use the same database. As in the previous study, we explained to the participants the dynamics of using the App and its operation. The time available to the participants of this study was five minutes in which it was suggested that they could use the App the way they wanted. An example of using this version can also be checked on the page created in `http://compomus.wixsite.com/compomus` [18].

Figure 5: Participants are making use of the App and verifying the movement of the sound.

After each round of use of the Applications, respondents were asked to answer the questionnaire prepared according to their vision and the feelings experienced during the study, were also asked to be as sincere as possible in their answers.

7.3 Result analysis

At this stage, the predetermined hypotheses were verified at the beginning of this section, relating them to the answers obtained in the questionnaire applied to the study participants. This analysis consisted of *in-loco* observations by the researchers and the documentation by audio and video recordings as well as the application of a post-test user experience questionnaire to record participants' impressions. Two graphs Figure 6 and Figure 7 show the distribution of study participant responses for each version. A more detailed analysis can be given below.

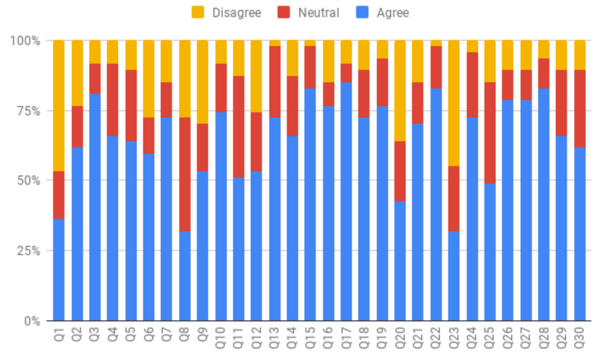


Figure 6: Average answers to the 30 questions in the JoyStick Version questionnaire.

H1: Considering questions 1 to 7, this hypothesis was confirmed, according to the figures about 53.19% of participants who used the PlayStop version pointed out in item 1 that there was a spontaneous organization by the group. On the other hand, 36.17% of the participants stated that there was no organization when they used the JoyStick version, 17.02% were undecided. In answering question 2, in which 68.09% of participants noted that the sound is most interesting when several people reproduce their sounds together in the PlayStop version. Users of the JoyStick version at about 61.7% also agree. [1] In response to question 4, 65.96% of the participants agreed that there was no type of prior organization for the generation generated in the use of the JoyStick version. This number was 76.60% among users who made use of the PlayStop version. When asked if they felt contributors to a composition in question 7, 70.21% of the participants agreed with the statement and felt responsible for the work generated in the PlayStop version. Already using the JoyStick version, users felt much more active in the participant part of the composition about 72.34% of users.

H2: Considering questions 8 to 13, this hypothesis is partially confirmed according to the numbers obtained in the answers to question 11. In this question, 51.06% of participants who used the JoyStick version stated that they have an affinity with electronic music since the sounds used are of this style. Already 59.57% of participants who used the PlayStop version also agreed to have a good relationship for style. Identification with the style of electronic music present in the App sounds caused an unexpected positive result among participants in their responses. This result does not fully confirm the hypothesis since a larger number of users with no affinity with the electronic music used for this study were expected. In answering question 8, 46.81% of the participants who used the PlayStop version were undecided when asked if the people in their circle of friends like the same musical styles. This number was 40.43% between JoyStick

[1] The results presented are the sum of the answers I agree to and strongly agree, as well as strongly disagree and strongly disagree.

version users. In answering question 10, about 80.85% of participants who made use of the PlayStop version like different/alternative ways of making music. Already 74.47% of users who made use of the JoyStick version also claim to like different/alternative ways of making music. Identification with the style of electronic music present in the App sounds caused an unexpected positive result among participants in their responses. This result does not fully confirm the hypothesis since a larger number of users with no affinity with the electronic music used for this study were expected.

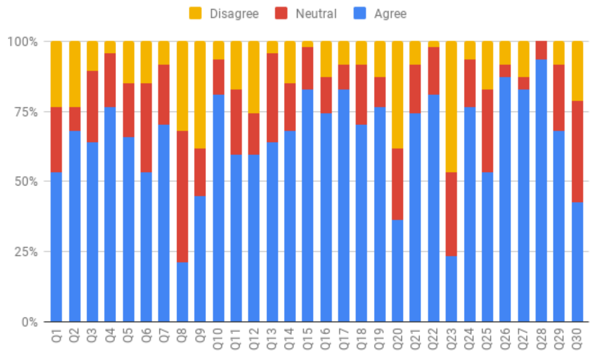


Figure 7: Average answers to the 30 questions in the PlayStop Version questionnaire.

H3: Considering questions 14 to 19, this hypothesis is confirmed. According to the figures found in the questionnaires when analyzing the answers to question 14, about 68.09% of users that were used the PlayStop version stated that they did not have experience with musical composition, but they not find difficulties to make music in this case. 65.96% of users who made use of the JoyStick version also agree that it was easy to make music even without experience in musical composition. In response to question 15, 82.98% of users who made use of the PlayStop version stated that the simplicity of the App made their experience with technology more accessible, this number was 82,98% between JoyStick users. In question 18, 70.21% of users who made use of the PlayStop version confirmed that they managed to dominate without any problems the dynamics of the use of the App. In the JoyStick version, this number was 72.34%.

H4: Considering questions 20 to 25, this hypothesis is confirmed according to the verified answers. In response to question 21, 70.21% of users of the JoyStick version stated that random sounds could indeed result in a single composition, among users of the PlayStop version that number was 74.47%. In question 22, 82.98% of JoyStick users concluded that this is a musical expression type. Among the users of the PlayStop version, this number was 80.85%. In response to question 24, 72.34% of users of the JoyStick version have confirmed they have enjoyed creating music from different sounds. 76.60% users of the PlayStop version also confirmed that they enjoyed creating music in this way.

H5: Considering questions 26 to 30, this hypothesis was confirmed according to the analysis of the answers collected. In response to question 26, 78.72% of users of the JoyStick version confirmed to use more than one electronic device daily. This number was 87.23% of the PlayStop version users. In response to question 28, 82.98% of participants who used the JoyStick version claimed to have great skill in the handling of electronic devices. Already 93.62% of users who tested the PlayStop version claim to have great skills when it comes to electronic devices. In question 29, 65.96% of users who used the JoyStick version claimed to control the sound in a proposed way without much trouble. This number among users of the PlayStop version was 68.09%.

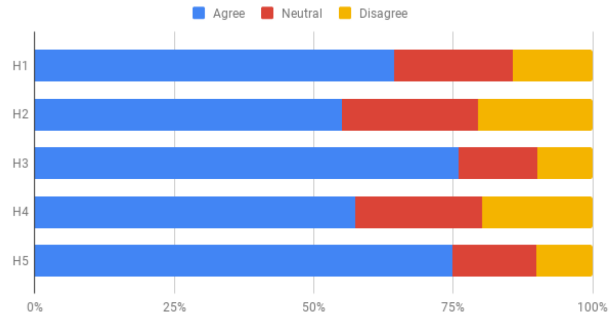


Figure 8: Results from PlayStop TAM questionnaire (average by hypothesis).

Overall, the results achieved in terms of technology acceptance are positive. The hypotheses are considered confirmed when they reach 50% or more in the average of answers that agree with the applied questions Figures 8 and 9.

The interaction with sound through mobile technologies is an artifact with the potential to be explored in the public art field. The interaction of the participants with the spatialization of sound is also another factor to be investigated since it gives new creative possibilities to the composers.

This study evaluated the potentialities and alternatives that can best be employed in the next version based on what was analyzed through the applied questionnaires. The compositions generated from the interaction of the

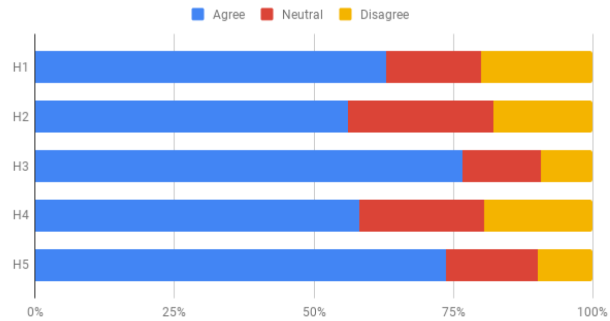


Figure 9: Results from JoyStick TAM questionnaire (average by hypothesis)

participants can be considered unique and irreproducible since they are the factor that inserts the randomness and unpredictability in work. This leads us to a random collaborative music composition using a methodology different from that used by John Cage. In this scenario, the agent of change is the very interaction of the participants with the sound through their smartphones (" instruments") in space either by reproducing the sound or by working their spatiality.

8 Concluding Remarks

We presented the Compomus app, a framework for collaborative music composition for immersive interaction with sound. We defined a case study with two versions of the Compomus: JoyStick and PlayStop.

The participating subjects in the study interacted with the sound in a university place simulating a public space by their mobile devices (smartphones). Participants have used both versions of Compomus, the first version, PlayStop, detects when the user is present in the environment, to play their sound. And JoyStick, in this version, the user besides the control of the sound reproduction also has the power of the sound spatialization and can direct its sound in any direction. At the end of the interactions, the users answered a questionnaire based on five hypotheses that aimed to evaluate the acceptance of the technology. Thus, four of the five assumptions were confirmed, and one of them was partially established, showing empirically in the data a good acceptance by the users. As a result, ten sound compositions were generated. Five compositions were obtained through the PlayStop version and are available on the stereo system. The other five compositions were obtained through the JoyStick version in the quadraphonic system.

We make available on a website [18] to participants and evaluators of this study could verify the results of the interactions for the two Compomus app versions proposed.

References

[1] Vicky Cattell, Nick Dines, Wil Gesler, and Sarah Curtis. Mingling, observing, and lingering: Everyday public spaces and their implications for well-being and social relations. *Health & Place*, 14(3):544 – 561, 2008.

[2] Tao Jiang. Urban public art and interaction design strategy based on digital technology. *Cluster Computing*, 0123456789:1–8, 2018.

[3] Camilo Salazar. What is sound art? `https://www.morleycollege.ac.uk/news/160-what-is-sound-art`, 2019. [Online; accessed 29-May-2019].

[4] Aleksandar Matic, Venet Osmani, and Oscar Mayora-Ibarra. Analysis of social interactions through mobile phones. *Mobile Networks and Applications*, 17(6):808–819, 2012.

[5] Trident Cloud Limited. Cloud recording sound to band. `https://itunes.apple.com/us/app/trackd-music-social-studio/id978196692`, 2019. [Online; accessed in 26-April-2019].

[6] Rossi. Nathália Angela Fragoso and Barbosa. Rogério Vasconcelos. A música indeterminada de john cage: a relação compositor-intérprete. *XXV Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, 2015.

[7] Eloy Fernando Fritsch. *Música Eletrônica-Uma Introdução Ilustrada*. Eloy Fernando Fritsch, 2008.

[8] Davide Rocchesso, Stefania Serafin, Frauke Behrendt, Nicola Bernardini, Roberto Bresin, Gerhard Eckel, Karmen Franinovic, Thomas Hermann, Sandra Pauletto, Patrick Susini, and Yon Visell. Sonic interaction design: Sound, information and experience. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 3969–3972, New York, NY, USA, 2008. ACM.

[9] Thomas Hermann, Andy Hunt, and John G Neuhoff. *The sonification handbook*. Logos Verlag Berlin, 2011.

[10] Jorge Lopes Elen Nas. Design De Interfaces Para Experiências Lúdicas Com Novas Tecnologias: O Caso Do Software-Livre Reactivision Em Interações Sonoras. *12º Congresso Brasileiro de Pesquisa e Desenvolvimento em Design*, 2016.

[11] Pro Tools Cloud Collaboration. Link play together. `https://www.avid.com/pro-tools/cloud-collaboration`, 2019. [Online; accessed in 26-April-2019].

[12] Siqueira. Denise da Costa Oliveira. Os jovens e suas telas pós-modernas: reflexões em torno da cidade, do imaginário e da tecnologia. *Comunicação e Sociedade*, 2010.

[13] B. Moore. An introduction to the psychology of hearing. 5 ed. Academic Press., 2003.

[14] M. Gerzon. Surround-sound psychoacoustics. Wireless World, p. 483-485, dez. 1974b., 1974.

[15] Denise Hortência Lopes Garcia. Modelos perceptivos na música eletroacústica. *São Paulo, PUC [Doutorado em Comunicação e Semiótica]*, 1998.

[16] Regis Rossi A Faria, Leandro F Thomaz, Luciano Soares, Breno T Santos, Marcelo K Zuffo, and João Antônio Zuffo. Audience-audio immersion experiences in the caverna digital. In *Proceedings of the10th Brazilian Symposium on Computer Music, Belo Horizonte*, pages 106–117, 2005.

[17] El País Brasil. Android já é o sistema operacional mais usado do mundo. `https://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467_396232.html`, 2019. [Online; accessed in 12-May-2019].

[18] Compomus. Media generated from study with compomus playstop and joystick. `http://compomus.wixsite.com/compomus`, 2019. [Online; accessed 10-May-2019].

[19] LANDR. Free sample packs to inspire your next project. `https://samples.landr.com/`, 2019. [Online; accessed in 26-April-2019].

[20] Robert K Yin. *Estudo de Caso: Planejamento e Métodos*. Bookman editora, 2015.

[21] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw. User acceptance of computer technology: a comparison of two theoretical models. *Comunicação e Sociedade*, 1989.

[22] Marlon Dalmoro and Kelmara Mendes Vieira. Dilemas na construção de escalas tipo likert: o número de itens e a disposição influenciam nos resultados? *Revista gestão organizacional*, 6(3), 2014.

State of art of real-time singing voice synthesis

Leonardo Araujo Zoehler Brum[1], Edward David Moreno[1]

[1]Programa de Pós-graduação em Ciência da Computação – Universidade Federal de Sergipe
Av. Marechal Rondon, s/n, Jardim Rosa Elze – 49100-000 São Cristóvão, SE

leonardo.brum@dcomp.ufs.br, edwdavid@gmail.com

Abstract

This paper describes the state of art of real-time singing voice synthesis and presents its concept, applications and technical aspects. A technological mapping and a literature review are made in order to indicate the latest developments in this area. We made a brief comparative analysis among the selected works. Finally, we have discussed challenges and future research problems.

Keywords: Real-time singing voice synthesis, Sound Synthesis, TTS, MIDI, Computer Music.

1. Introduction

The aim of singing voice synthesis is to computationally generate a song, given its musical notes and lyrics [1]. Hence, it is a branch of text-to-speech (TTS) technology [2] with the application of some techniques of musical sound synthesis.

An example of application of singing voice synthesizers is in the educational area. Digital files containing the singing voice can be easily created, shared and modified in order to facilitate the learning process, which dispenses the human presence of a singer as reference, or even recordings.

This kind of synthesis can also be used for artistic purposes [3]. Investments on the "career" of virtual singers, like Hatsue Miku, in Japan, have been made, which includes live shows where the singing voice is generated by Vocaloid synthesizer, from Yamaha, and the singer image is projected by holograms.

The singing voice synthesis technology applications have been increased by the development of real-time synthesizers, like Vocaloid Keyboard [4], whose virtual singer is implemented by an embedded system into a keytar, allowing its user the execution of an instrumental performance.

The present article presents a review about real-time singing voice synthesis embedded systems, through the description of its concept, theoretical premises, main used techniques, latest developments and challenges for future research.

This work is organized as follows: Section 2 describes the theoretical requisites which serve as base for singing voice synthesis in general; Section 3 presents a technological mapping of the patents registered for this area; in Section 4 the systematic review of literature is shown; Section 5 contains a comparative analysis among the selected works; Section 6 discuss the challenges and future tendencies for this field; finally, Section 7 presents a brief conclusion.

2. Theoretical framework

Singing voice synthesis has two elements as input data: the lyrics of the song which will be synthesized and musical parameters that indicate sound qualities. The lyrics can be inserted according to the orthography of the respective idiom or through some phonetical notation, like SAMPA, while the musical parameters can be given by MIDI messages or other file formats, such as MusicXML. The expected output is a digital audio file which contains the specified chant.

In respect of data processing, the main singing voice synthesis techniques consist in a combination of text-to-speech (TTS) and musical synthesis. In the early years of this area, two approaches were developed: rule-based synthesis, which computationally generates sound according to its physical characteristics and sample-based syntheses, that uses pre-recorded audio. The data-driven approach has been developed recently. It uses statistical models.

Singing voice synthesis problem domain is, therefore, multidisciplinary: beyond computer science, it depends on concepts from acoustics, phonetics, musical theory and signal processing. The following subsections present how each mentioned knowledge area interact in the main singing synthesis technical approaches, with a brief description of real-time synthesis at the end of this article.

2.1. Rule-based approaches

Rule-based singing synthesis considers the way sound is produced, by the analysis of its physical characteristics, which are applied in the artificially generated signal.

Sound is a physical phenom generated by the variation, through time, of atmospheric pressure levels provided by a vibratory source. Given its wavy nature, sound has physical quantities such as period, frequency and amplitude. **Period** consist in the duration of a complete wave cycle. **Frequency** is the inverse of period and indicates how many cycles per second has the sound wave. Finally, **amplitude** is the maximum value of pressure variation in relation to the equilibrium point of the oscillation [8]. The period and amplitude of a simple sound wave can be seen in the Figure1.

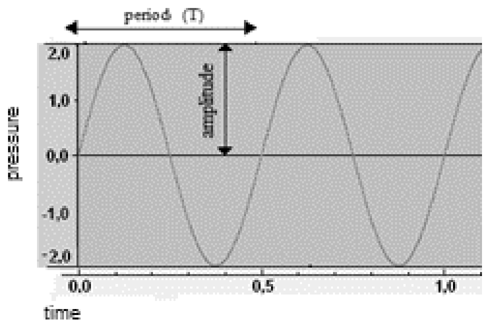


Figure 1. Simple sound wave.

The simplest sound waves are called sinusoids and have a single frequency. However, this kind of sound is not produced neither by nature, nor by the conventional musical instruments. Such sources generate **complex sounds**, composed by several frequencies. The lowest frequency in a complex sound is called **fundamental frequency**. In case of the other frequency values of the sound be multiples of the fundamental frequency, the sound is said to be **periodic**. If not, the sound is called **aperiodic**. The superposition of frequencies results in the **waveform** of each sound source and describes a shape called **envelope**, obtained from the maximum oscillation values of the waveform.

The envelope shape is commonly decomposed into four stages, indicated by the ADSR acronym: attack, which corresponds to the period between the beginning of the sound execution and its maximum amplitude; decay, the time needed for the sound depart form is maximum amplitude towards a constant one; sustain, the interval in which such constant state persists; release, whose duration is between the constant state and the return to silence.

In respect of speech, specifically, its smallest distinguishable unit is called **phoneme**. Phonemes are classified into two main groups: **consonants** and **vowels**. From an acoustic perspective, both groups consist in a set of complex sounds: consonants are aperiodic vibrations that result from an obstruction of the air flow by body parts such as lips or the tongue. On the other hand, vowels have periodic nature.

Another factor of differentiation between vowels and consonants is the role that each type of phoneme performs in the syllable. According to the frame/content theory [5], speech is organized in syllabic frames which consist in cycles of opening and closure of mouth. In each frame there is a segmental content, the phonemes. This content has three structures: attack, nucleus and coda. Consonantal phonemes are located in attack or coda, while a vowel form the syllable nucleus [6]. Beyond consonants, there are phonemes called semivowels, which form diphthongs and can also appear in attack or coda. Acoustically speaking, syllable is a waveform in which consonants and semivowels are found in attack and release (coda) stages, while vowels are in sustain (nucleus) stage of the envelope shape. Figure 2 schematically presents the relation between the structure of the Brazilian Portuguese syllable “pai” and the envelope shape stages.

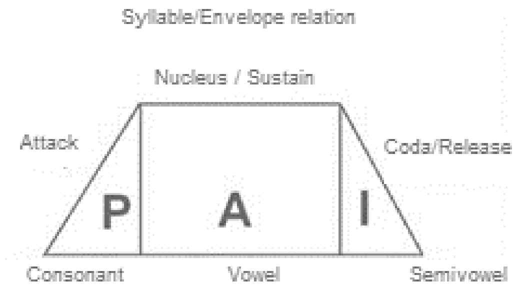


Figure 2. Relation between a syllable structure and the envelope shape stages.

Furthermore, vowels have another important feature, caused by resonances produced by the human vocal apparatus. These elements are called **formants** and appear as energy peaks verified when an acoustic signal is analyzed from its spectrum, in frequency domain. The spectral representation put in evidence the most relevant frequencies of a complex sound in relation to amplitude. Formants have the value of the central frequency in each energy peak that appears in the shape called spectral envelope. They are commonly enumerated from the lowest frequencies, as F1, F2, F3 and so on [7]. Figure 3 presents the spectrum of a voice signal where four formants are indicated.

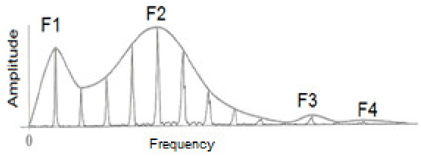


Figure 3. Spectral envelope of a voice signal with the indication of four formants.[7]

The presence of formants allows one to perceive the difference between vowels, since the sound qualities, such as pitch and loudness, of “a” and “o”, for example, can be identical.

Formant synthesis is an example of rule-based singing voice synthesis approach, which consists in the generation of units called *Forme d’Onde Formantique* (FOF, French for formant waveform). FOFs are sinusoids with a very short duration whose frequencies are equal to the value of the formants of the phoneme to be synthesized. Each FOF is then repeated according to the periodicity of the fundamental frequency of the musical note that is intended to synthesize. This process produces a series of sequences which are summed in order to generate the synthesized singing voice [7].

Systems based on this kind of approach, such as CHANT, developed by *Institut de Recherche et Coordination Acoustique/Musique* (IRCAM) at the early 1980’s, are among the first ones in respect to the use of synthesized voices for artistic purposes. They are capable of synthesize realistic vowels, but it costs a big studio effort to analyze and adjust parameters [2].

2.2. Sample-based approaches

In regard of perception, by human audition, of the sound phenom, the physical quantities previously mentioned are related to the so-called **qualities of sound**: **pitch**, that allows humans to distinguish between high and low sounds and is proportional to fundamental frequency; **loudness**, which depends on amplitude and indicate the difference between loud and soft sounds; timbre, related to the waveform, and for consequence, to the envelope shape, is the quality perceived as the proper “voice” of each sound source, which permits to distinguish, for example, the sound of piano from the guitar one, even if both have the same pitch and loudness. A fourth quality which can be cited is the **duration** of sounds [8].

Periodic complex sounds are usually perceived as having a defined pitch, which corresponds to the fundamental frequency. These sounds are called **musical sounds**. On the other hand, aperiodic sounds, which do not have a clearly distinguishable pitch, are denominated as **noises**, although they also are employed in music, specially through percussion instruments.

Since the development of musical theory in Western World, certain ways to select and organize sounds for artistic purposes and to graphically represent them according to its qualities, were created. The sensation of likeness when two sounds are heard and one of them has two times the fundamental frequency of the other, allowed the division of the audible frequencies range into musical scales, which consist in a set of individual sounds called **musical notes** [8]. The succession of sounds with different pitches or, in other words, the succession of different musical notes is the part of music denominated **melody**.

Musical sounds can be synthesized by means of sample-based approach, where recordings form real musical instruments are stored and handled according to the needs of the musician. From the recorded samples, other musical notes whose pitch is near are generated, while timing is treated as follows: if the note duration is less than the sample duration, the execution is interrupted; if not, there are two possibilities that depend on the instrument which is intended to be synthesized. For some instruments, like piano, the recorded sample will be executed till its end, returning to silence; for other instruments, like organ or flute, it is wanted for its execution to be prolonged as the note stays activated, either by a keyboard or some software. This indefinite prolongation is a result of the application of the looping technique, where a specific part of the sustain stage of the waveform is continuously repeated. The end of the note activation makes execution gets out of the loop towards the release stage [9].

One of the technologies widely employed to perform sample-based synthesis is MIDI (acronym for Music Interface Digital Instrument) protocol, developed by Yamaha at the 1980’s. This protocol provides communications between electronic musical instruments and computers. Its messages and file format do not contain audio signals, but only musical parameters that correspond to the sound qualities: musical note/pitch, dynamics/loudness, musical instrument/timbre, duration, among others. Such parameters serve as base for an on-demand handling of the sound samples, which can be stored in a computer or even in a musical instrument.

Sample-based synthesis can be also applied in order to perform the conversion of text into speech (TTS, text-to-speech). The greatest challenge of this approach is that how much larger is the size of the samples, more natural the result will sound, but the range of possible expressions will be smaller. Thus, for example, in case of developing a speech synthesizer for Brazilian Portuguese language taking word by word as samples, it will demand hundreds of thousands of recordings. On the other hand, if samples were at phoneme level, they would be less than a hundred and could generate any word, but the result of their concatenation will sound invariably “robotic” [2].

The challenges of sample-based TTS technique are naturally transposed for singing synthesis field, in case of performing it through the same approach. The song lyrics associated to a melody serves as input data, where each syllable corresponds to a single musical note. The phonetic samples are then being concatenates as the input is read by the system.

The looping technique, previously described, is applied on vowels, because they are periodic, musical sounds which correspond to the musical notes and to the sustain stage of each syllable. This process prolongs the syllable duration according to the musical parameters of the input. Consonants and semivowels are concatenated at the vowel’s margins [9].

Pre-recorded samples are commonly stored in a singing library which consists in units that can be modeled to contain one or more phonemes. In singing voice, the pitch variation among vowels is much less than in speech, because the first one is driven by the musical notes, but this fact does not exclude the difficulties to obtain a “realistic” result from samples in singing synthesis [10].

An example of system that performs concatenative singing voice synthesis is Vocaloid [3], which has achieved great commercial success. Vocaloid has a piano roll-type interface, composed by a virtual keyboard associated to a table whose filling is correspondent to the chosen musical notes. Input can be made by means of conventional peripherals, such a mouse, or through electronic musical instruments that support MIDI protocol. The song lyrics is associated to musical notes as it is typed into the piano roll. Input data is sent to the synthesis engine, serving as reference to the selection of samples stored in the singing library. A system diagram of Vocaloid is shown by Figure 4.

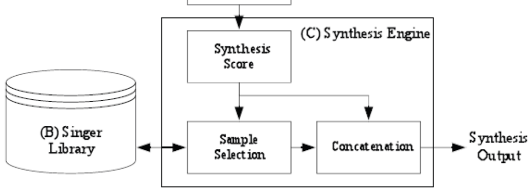


Figure 4. System diagram of Vocaloid [3].

2.3. Data-driven approaches

In the last years, some singing synthesizers have been developed based on probabilistic models, which differ from the deterministic nature of the rule-based approach. Tools like Hidden Markov Model (HMM) [1], successfully employed in TTS systems, are useful, for example, to apply in samples that contains a single phoneme the behavior provided by a statistical analysis of the voice of a singer. This decreases the size of the singing library and minimizes the lack of “naturalness” of the synthesizes voice in a more efficient way than the concatenative approach. The adjust of parameters performed by this kind of model is commonly called **training**, while the signal from which the parameters are extracted is denominated **target**.

The first HMM-based singing synthesizer was SinSy [11], developed by Nagoya Institute of Technology. This system is available on a website, where the upload of a MusicXML file, a format generated by most of the music score editors, can be made as input. SinSy provides as output a WAV file that contains the synthesized singing voice. The idioms supported by SinSy are English and Japanese.

2.4. Real-time singing voice synthesis

Users of synthesizers like Vocaloid define input data (song lyrics and musical notes) for the system in order to generate the singing voice later, in such a way analog to an IDE engine, where design time and run time are distinct.

This limitation has been overcome by the development of real-time singing voice synthesizers. They are embedded systems that artificially produce chant at the very moment the input data is provided by the users, which allows to use the synthesizer as a musical instrument [12].

In order to achieve a better comprehension of this new branch of singing synthesis, the present work performed a scientific mapping, according to the methodology proposed in [13]. The research questions that must be answered are the following ones: (i) What are the singing synthesis techniques employed by most of the real-time systems? and (ii) What are the input methods that such systems use in order to provide the phonetic and musical parameters?

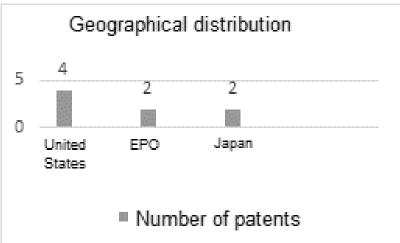
The scientific mapping consists in a technological mapping, where the patents related to real-time singing synthesis are searched, and a literature review. Both parts of the scientific mapping are described by the next two sections.

3. Technological mapping

The search for patents related to real-time singing voice systems was performed in two different databases: WIPO (World Intellectual Property Organization) and INPI (Instituto Nacional da Propriedade Industrial, from Brazil). The INPI database returned no results, even for more generic search keys in English and Portuguese, like “síntese de voz cantada” or “singing synthesis”. The WIPO database, for its turn, provided some patent deposits from the following search string:

FP:(FP:((" SINGING SYNTHESIS " OR "SINGING VOICE SYNTHESIS" OR "SINGING SYNTHESIZING") AND ("REAL TIME" OR "REAL-TIME")))

The research presented eight records as result. All of them were property of Yamaha Corporation, from Japan, and their author was Hiraku Kayama, except by one, whose author was Hiroshi Kayama. However, most of the patents were registered outside Japan, probably in order to warrant international legal protection. Graphic 1 presents the geographical distribution of the patents, where EPO is the European Patent Office.



Graphic 1. Patents geographical distribution.

All patents had as object a method, apparatus and storage medium for real-time singing voice synthesis. It is a product, developed by Yamaha, which consists in a musical keyboard with an embedded singing synthesizer, allowing the user to make an instrumental performance with its virtual singer. The product was denominated Vocaloid Keyboard [4].

A prototype of the instrument, presented in 2012, had alphabetic buttons at left, organized as follows: two horizontal rows with consonants and diacritical signs and, bellow them, five buttons with vowels, organized in a cross shape. With left hand, the user could activate these buttons to generate syllables, meanwhile the musical keyboard could be played by the right hand to indicate the musical notes. The generated syllables were shown in a display with *katakana* Japanese characters. Figure 5 shows the described prototype.

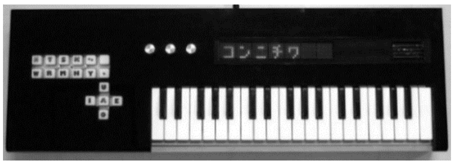


Figure 5. Vocaloid Keyboard prototype [4].

This device was designed to synthesize singing in Japanese and the phonetic limitations of such idiom favored this kind of interface. The prevalent structure of Japanese syllables is consonant-vowel, which means that, for example, when “S” and “A” buttons are simultaneously activated, the systems generates the “SA” syllable, since a syllabic structure like “AS” does not exist in Japanese [14].

The singing synthesis technique employed was the concatenative one, the same of the Vocaloid software, and the instrument is already in commercialization. In respect to hardware, an Arduino board was one of the used technologies, at least in the prototyping phase [4].

4. Systematic review

The systematic review of literature consisted, in first place, in a search performed on the Scopus scientific database, with the following search string:

TITLE-ABS-KEY (("singing synthesis " OR "singing voice synthesis") AND ("REAL TIME" OR "REAL-TIME"))

This search returned nineteen records, and the works were selected according to the following criteria: (i) The work must have been published in the last ten years; (ii) The work must describe a new product.

Six works were excluded by the chronologic criterion; two of them did not describe a new product, but evaluation methods; finally, other three records were excluded because they were repeated. Searches were made in other scientific databases, like IEEE Xplore and ACM, but they did not return different results. Hence, eight articles were selected and evaluated. A brief description of each one of them follows.

FEUGÈRE *et al.* (2017) [15] present a system called Cantor Digitalis, whose input method in denominated chironomy and consists in an analogy between hand movements and the phonetic and musical parameters required by singing synthesis. The system performs formant synthesis, which produces only vowels. With one of the hands, the user touches a tablet with a stylus, in order to indicate the wanted melodic line; simultaneously, the vowel to be synthesized is indicated by gestures made by the fingers of the other hand on the tablet.

LE BEUX *et al.* (2011) [16] proposed the integration of several instances of Cantor Digitalis by means of an environment called Méta-Mallete, which allows to execute simultaneously several computational musical instruments on the same computer through USB interfaces. Since several singing synthesizers could be used at the same time, it was possible to present a virtual choir, which was denominated Chorus Digitalis.

DELALEZ and D'ALESSANDRO (2017) [6] used the interface of Cantor Digitalis and connected it to pedals in order to build another system, called VOKinesiS, which transforms pre-recorded voice samples by means of a pitch control provided by Cantor Digitalis, while the pedals indicate timing parameters that change the rhythm of the original samples.

The work of CHAN et al. (2016) [12] describes the development of a real-time synthesizer called SERAPHIM that intends to overcome certain limitations of Cantor Digiatlis — which produces only vowels — and of Vocaloid Keyboard, whose real-time synthesis capabilities are at frame (syllable) level, but not at content (phoneme) level. SERAPHIM system provides a gestural input that allows to synthesize phoneme by phoneme, either vowels or consonants, in real time. The technique employed is sample-based concatenative synthesis, with a singing library stored in indexed structures denominated wavetables.

The I²R Speech2Singing system, developed by DONG et al. (2014) [17], instantly converts a voice input into singing, through the application of characteristics of the voices of professional singers, stored in its database, over the user voice. Hence, this system employs a data-driven approach, where the parameters are extracted from a signal and applied into another one.

MORISE *et al.* (2009) [18] developed an interface denominated v.morish'09, which also provides the transformation of a voice signal that serves as input according to characteristics extracted from a professional singer's voice.

The synthesizer of GU and LIAO (2011) [19] is a system embedded in a robot designed to present singing abilities. The system uses the harmonic plus noise model (HNM) in order to adjust parameters. The pre-recorded 408 syllables of Mandarin Chinese language serve as target signals.

YU (2017) [20] uses data-driven approach with HMM in order to develop his synthesizer, with an additional feature: the system it is integrated to a 3D animation which articulates mouth movements.

In the last two works the musical parameters are provided by a static file which contains musical notes. The real-time nature of these systems is related to the operations of the robot and the 3D animation.

In next section, a brief comparative analysis among the selected works will be made in order to answer the proposed research questions.

5. Comparative analysis

The research questions of this work were presented in Section 2. The first of then was "What are the singing synthesis techniques employed by most of the real-time systems?".

To answer it, the following comparative chart present the technical approaches used by real-time singing voice synthesizers described by each one of the works selected in the systematic review, with the addition of Vocaloid Keyboard, which was the result of the technological mapping, having as reference the paper of KAGAMI *et al.* (2012) [4]. The articles appear in chronological order.

Article	**Rule-based approaches**	**Sample-based approaches**	**Data-driven approaches**
MORISE *et al.* (2009) [18]			✓
GU; LIAO (2011) [19]			✓
LE BEUX *et al.* (2011) [16]	✓		
KAGAMI *et al.* (2012) [4]		✓	
DONG *et al.* (2014) [17]			✓
CHAN *et al.* (2016) [12]		✓	
DELALEZ (2017) [6]		✓	
FEUGÈRE (2017) [15]	✓		
YU (2017) [20]			✓

Chart 1. Technical approaches for real-time singing synthesis discussed by the selected works.

Among the nine evaluated works, four employed a data-driven approach; three used a sample-based one; finally, two of them used a rule-based approach. In a such restricted universe, it is possible to assert that all the main approaches of singing synthesis in general are relevant for the specific branch of real-time singing synthesis.

A geographical hypothesis could explain such equilibrium: works [16] and [15] were produced in European institutes that, under the influence of IRCAM, developed Cantor Digitalis synthesizer using the formant synthesis technique. The paper [6] also employed features of Cantor Digitalis, but in order to overcome the limitation of providing only vowels, it needed to use samples, so the Cantor Digitalis interfaces only served to control certain parameters.

In Asia, data-driven approach is prevalent, as works [18], [19], [17] and [12] indicate. For its turn, sample-based approach continues to be promoted by Yamaha, with the development of Vocaloid Keyboard [4]. The SERPHIM synthesizer [12] was developed using sample-based approach as it takes Vocaloid Keyboard as reference.

The other research question proposed by the present work was about the input methods employed by the synthesizers. It is a critical element in systems that provide a real-time performance. The selected works present four basic input types: static files, musical instruments, electronic devices (tablets, for example) and voice signals. Chart 2 present a comparison among the articles in relation to this aspect.

Article	Static files	Musical instruments	Electronic devices	Voice signal
MORISE *et al.* (2009) [18]				✓
GU; LIAO (2011) [19]	✓			
LE BEUX *et al.* (2011) [16]			✓	
KAGAMI *et al.* (2012) [4]		✓		
DONG *et al.* (2014) [17]			✓	✓
CHAN *et al.* (2016) [12]			✓	
DELALEZ (2017) [6]			✓	✓
FEUGÈRE (2017) [15]			✓	
YU (2017) [20]	✓			

Chart 2. Input method used by the singing synthesizers described in the selected works.

The option for static files was made by systems where the synthesized singing voice worked as a real-time controller of other elements: a robot in [19] and a 3D facial animation in [20].

In works [18], [17] e [6], a voice signal acts as input in order to provide simultaneously the phonetic and musical parameters required for singing synthesis. The systems presented by these works provide as output a synthesized voice that change or “correct” the musical imperfections of the input.

The only work whose interface consisted in a conventional musical instrument was [4], because of the nature of the proposed commercial product. It is important to remark that the combination between the musical keyboard and the textual buttons was possible because of the phonetic limitations of Japanese idiom, for which this synthesizer was designed.

In more than a half of the works [16], [17], [12], [6], [15], other hardware devices were employed as input method.

6. Challenges and future works

The main challenge of singing voice synthesis in general is to achieve naturalness to the generated chant, because, beyond any subjective aspect, the adjust of parameters that provide such characteristic requires a more complex processing than the simple extraction of data from the musical input.

In the specific case of real-time singing synthesis, one of the most complex challenges is to provide an input method that conciliate phonetic and musical data simultaneously. The present work indicated that even a human voice signal has been used in order to perform this role. On the order hand, for specific idioms, like Japanese, a conventional musical interface was successfully adapted with buttons that provide phonetic parameters.

A future work, still inedited, would be the development of a real-time singing synthesizer for Brazilian Portuguese language. The input data could be provided by a static file with phonetic data, while a MIDI keyboard would be able to provide the musical parameters during a performance.

7. Conclusion

The field of real-time singing voice synthesis is still very restricted, with a small number of works developed in comparison to other areas where embedded systems are employed, such as IoT and neural networks. All the main approaches used by singing synthesis in general are also employed by the real-time synthesizers and several solutions are adopted in order to overcome the challenges that are inherent to the input methods.

References

[1] KHAN, Najeeb Ullah; LEE, Jung Chul. HMM Based Duration Control for Singing TTS. In: Advances in Computer Science and Ubiquitous Computing. Springer, Singapore, 2015. p. 137-143.

[2] ALIVIZATOU-BARAKOU, Marilena *et al.* Intangible cultural heritage and new technologies: challenges and opportunities for cultural preservation and development. In: Mixed Reality and Gamification for Cultural Heritage. Springer, Cham, 2017. p. 129-158.

[3] KENMOCHI, Hideki. Singing synthesis as a new musical instrument. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2012. p. 5385-5388.

[4] KAGAMI, Shota *et al*. Development of Realtime Japanese Vocal Keyboard. In: Information Processing Society of Japan INTERACTION, pages 837-842, 2012.

[5] MACNEILAGE, Peter. The frame/content theory of evolution of speech production. Behavioral and brain sciences, v. 21, n. 4, p. 499-511, 1998.

[6] DELALEZ, Samuel; D'ALESSANDRO, Christophe. Adjusting the frame: Biphasic performative control of speech rhythm. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pages 864-868, 2017.

[7] LOY, Gareth. Musimathics: the Mathematical Fundamentals of Music., MIT Press, 2011.

[8] BADER, Rolf (Ed.). Springer handbook of systematic musicology. Springer, 2018.

[9] BRUM, Leonardo Araujo Zoehler. Technical aspects of concatenation-based singing voice synthesis. Scientia Plena, v. 8, n. 3 (a), 2012.

[10] HOWARD, David. Virtual Choirs. In: The Routledge Companion to Music, Technology, and Education. Routledge, 2017. p. 305-314.

[11] OURA, Keiichiro *et al*. Recent development of the HMM-based singing voice synthesis system—Sinsy. In: Seventh ISCA Workshop on Speech Synthesis. 2010.

[12] CHAN, Paul Yaozhu, *et al*. SERAPHIM: A wavetable synthesis system with 3D lip animation for real-time speech and singing applications on mobile platforms. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pages 1225-1229, 2016.

[13] PETERSEN, Kai *et al*. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, v. 64, p. 1-18, 2015.

[14] KUBOZONO, Haruo (Ed.). Handbook of Japanese phonetics and phonology. Walter de Gruyter GmbH & Co KG, 2015.

[15] FEUGÈRE, Lionel *et al*. Cantor Digitalis: chironomic parametric synthesis of singing. EURASIP Journal on Audio, Speech, and Music Processing, v. 2017, n. 1, p. 2, 2017.

[16] LE BEUX, Sylvain *et al*. Chorus digitalis: Experiments in chironomic choir singing. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pages 2005-2008, 2011.

[17] DONG, Minghui *et al*. I²R speech2singing perfects everyone's singing. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH pages 2148-2149, 2014.

[18] MORISE, Masanori et al. v. morish'09: A morphing-based singing design interface for vocal melodies. In: International Conference on Entertainment Computing. Springer, Berlin, Heidelberg, 2009. p. 185-190.

[19] GU, Hung-Yan; LIAO, Huang-Liang. Mandarin singing voice synthesis using an HNM based scheme. In: 2008 Congress on Image and Signal Processing. IEEE, 2008. p. 347-351.

[20] YU, Jun. A Real-Time 3D Visual Singing Synthesis: From Appearance to Internal Articulators. In: International Conference on Multimedia Modeling. Springer, Cham, 2017. p. 53-64.

Visualizing Air Drums: Analysis of Motion and Vocalization Data Related to Playing Imaginary Drums

Ana Julia Pereira Caetano[1 2] **, Tiago Fernandes Tavares**[1 2]

[1]School of Electric and Computer Engineering (FEEC) – University of Campinas (Unicamp)
CEP 13083-852 – Av. Albert Einstein, 400 – Campinas – SP – Brazil

[2]Interdisciplinary Nucleus for Sound Studies (NICS) – University of Campinas (Unicamp)
CEP 13083-872 – Rua da Reitoria, 165 – Campinas – SP – Brazil

tavares@dca.fee.unicamp.br

Abstract. *Air drums, or imaginary drums, are commonly played as a form of participating in musical experiences. The gestures derived from playing air drums can be acquired using accelerometers and then mapped into sound control responses. Commonly, the mapping process relies on a peak-picking procedure that maps local maxima or minima to sound triggers. In this work, we analyzed accelerometer and audio data comprising the motion of subjects playing air drums while vocalizing their expected results. Our qualitative analysis revealed that each subject produced a different relationship between their motion and the vocalization. This suggests that using a fixed peak-picking procedure can be unreliable when designing accelerometer-controlled drum instruments. Moreover, user-specific personalization can be an important feature in this type of virtual instrument. This poses a new challenge for this field, which consists of quickly personalizing virtual drum interactions. We made our dataset available to foster future work in this subject.*

1 Introduction

Gesture-controlled virtual instruments can provide musicians an experience closer to that provided by acoustic instruments. This experience relies on gesture acquisition and instrument emulation [1]. However, not all acquisition or emulation methods can lead to musically meaningful instruments [2].

Musical meaningfulness can be pursued by a design process involving emulation of real, physical environments. For such, it is possible to use pattern recognition techniques. These techniques can be used to detect specific gestures, as well as their intensity and possible variations, and link them to sonic manifestations [3]. However, they require a reasonable amount of labeled data for parameter optimization [4].

Gesture-related labeled data is hard to obtain because it needs to be acquired from human subjects. Human acquired gesture data can account for gesture variations that are hard to predict with physical motion models. In addition, humans have particular prior experience and expectations regarding the behavior of virtual instruments [1].

This phenomenon has been studied by Maki-Patola [5], who designed an experiment in which subjects played virtual drum instruments using different interfaces. This experiment showed that playing precision varies according to people and interfaces. Maki-Patola used fixed tempo and predefined interfaces and interactions to emulate real acoustic drums.

Another approach to problem of emulating virtual drums was presented by Havel and Desainte-Catherine [6]. They proposed a virtual drum instrument specially designed for a specific musician. This instrument provides an interaction model that involves strike classification in addition to the detection. However, all collected data is related to one subject, therefore it cannot be generalized.

Another initiative towards the analysis of percussive gestures was performed by Dahl [7]. This study was based on a dataset consisted of free-hand movements acquired while subjects tried to synchronize to a pre-recorded rhythm. Dahl [7] studied the position, velocity and acceleration of the subjects' wrist and hand movements.

In this work, we present a dataset containing gestures data collected from 32 different subjects playing imaginary drums without accompanying music, which consists of a different condition that that analyzed by Dahl [7]. The dataset also contains vocalizations of the expected sonic results for each subject. The data acquisition process did not induce subjects to play in a particular *tempo*. Our dataset can be used for the construction of machine-learning based instruments that generalizes across different people.

We also performed data analysis showing that the alignment between the vocalization and its gesture signal is different for each subject. This difference can be observed regardless of their previous musical experience and rhythmic intention.

The remainder of this work is organized as follows. Section 2 describes the data acquisition process. Section 3 presents further analysis on the acquired data. Last, Section 5 concludes the paper.

2 Data Acquisition

Our data acquisition process relies on the assumption that different people expect different sound results when they play imaginary drums. With that in mind, we designed a data acquisition process in which subjects provide both gesture data and its respective expected vocalization.

The dataset contains data acquired from 32 subjects (23 male and 9 female), aged between 17 and 65 years old. Within this group, 20 subjects had previous musical experience and 12 did not (10 of them had experience in playing percursion, while 10 did not). All subjects are residents of the South-East of Brazil. They all signed a free consent form. This experiment was approved by the Ethics Committee of the University of Campinas (CAAE 53738316.0.0000.5404). Each subject was instructed to perform gestures that emulate playing an imaginary drum using a WiiMote as a stick, and vocalize the sound they imagine to produce. Each subject recorded two different tracks. In one of them, the subject was instructed to maintain a steady rhythm and *tempo*. In the other, they were instructed to perform free beat variations.

Audio was acquired using a laptop microphone and the WiiMote device data was upsampled to 44100 Hz. As a result, we generated 64 tracks containing time aligned gesture and vocalization. On average, each track is 8 seconds long. We only used the X axis of the WiiMote accelerometer because the acquired motions are more closely aligned to this axis.

In the next section we conduct further discussion about observed data.

3 Data Analysis

Our data analysis was based on observing the alignment between gestures and their corresponding vocalizations, as shown in Figure 1. As will be further discussed, this alignment varies, which indicates that different subjects imagined different interactions with their imaginary drums. We also observed the percussive gestures shape variations across subjects and acquisition conditions.

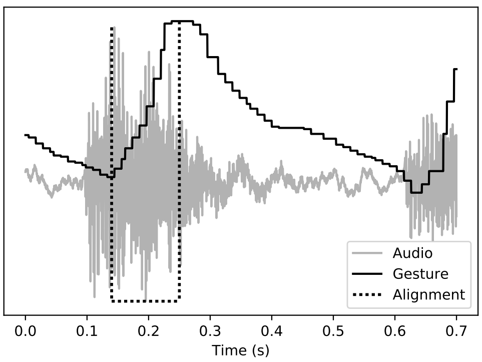


Figure 1: Audio and gesture alignment. It is possible to observe that the peak gesture activity happened after the percussive vocalization. This behavior was not consistent among the acquisitions.

Differences were observed regardless of the subjects' previous experience with percussive instruments. This aspect is further discussed in section 3.1.

We also observed that the relationship between gesture and its imagined sound changes for the same person according to their rhythmic intention. This means that performing different rhythms impacts on this relationship. A deeper discussion on this subject is conducted in section 3.2.

We selected data from specific subjects as shown in Table 1. The same subjects were used in the analyzes conducted in sections 3.1 and 3.2.

Table 1: Subjects selected for data analysis.

Subject	Experience
S1	No musical experience
S2	Non-percussive instrument experience
S3	Percussive instrument experience
S4	Percussive instrument experience
S5	Percussive instrument experience
S6	Percussive instrument experience

3.1 Impact of Previous Percussion Experience

Figures 2 and 3 show audio and gesture captured from subjects with no previous experience in percussion. Data shown in Figure 3 relates to a subject with experience in non percussion instruments. It is clear that the peaks and valleys related to the performed gesture and vocalization align differently for each subject. Moreover, the musically inexperienced subject (S1) presents less consistency in this alignment than the experienced one (S2).

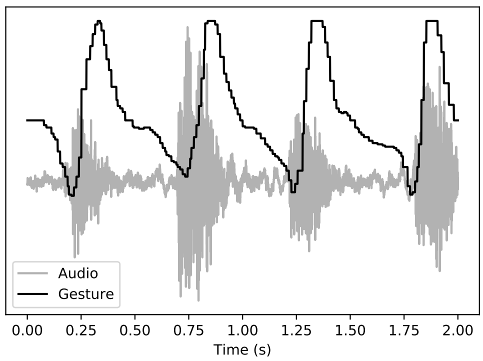


Figure 2: Audio and gesture related to a steady rhythm performed by S1.

Figure 4 shows the data acquired from S2, a subject with previous experience in percussion instruments. It is possible to observe that gesture and vocalization are aligned at their onsets and a valley in the gesture preceeds the vocalization. Figure 3 shows that this alignment can also be observed for S3. However, it is possible to see that S3 produces a sequence of two valleys before the vocalization, and two peaks after the vocalization, while S2 produces a single valley and a much smaller second peak. Also, the vocalization of S3 is closer to its preceding valley when compared to the vocalization of S2. This suggests that the imagined interaction is different for each one of them.

Figures 5, 6 and 7 depict the data captured from three other percussionists (respectively, S4, S5, and S6)

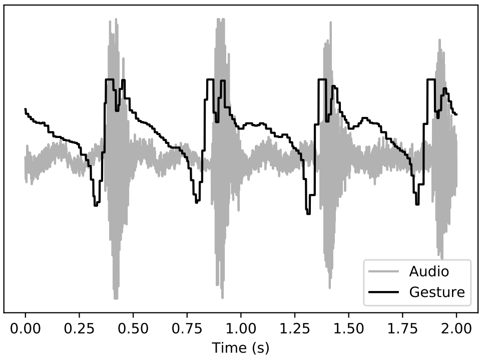


Figure 3: Audio and gesture related to a steady rhythm performed by S2.

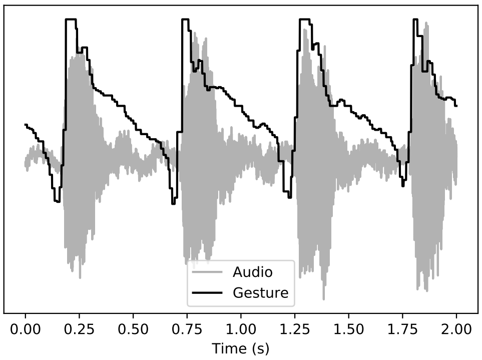


Figure 4: Audio and gesture related to a steady rhythm performed by S3.

with different levels of expertise. It is clear that the alignment of motion peaks and valleys with the vocalization is different. This difference is similar to that found between S2 and S3, which indicates that these differences are due to imagining different interactions or situations, not to differences in musical expertise.

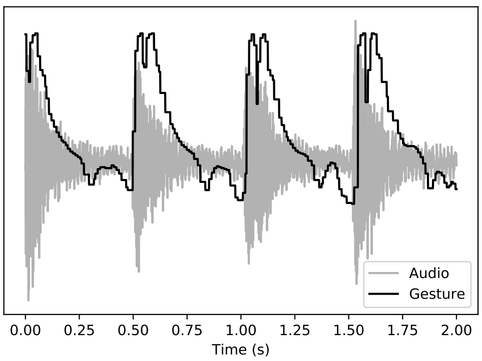


Figure 5: Audio and gesture related to a steady rhythm performed by S4.

Interestingly, data from S5 also shows alignment between the vocalization and the gesture activity valley, but this cannot be observed in S6 or S3. Also, the alignment

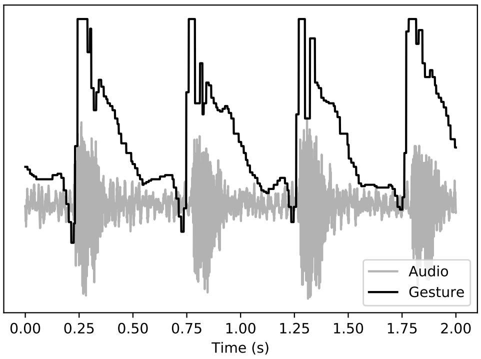


Figure 6: Audio and gesture related to a steady rhythm performed by S5.

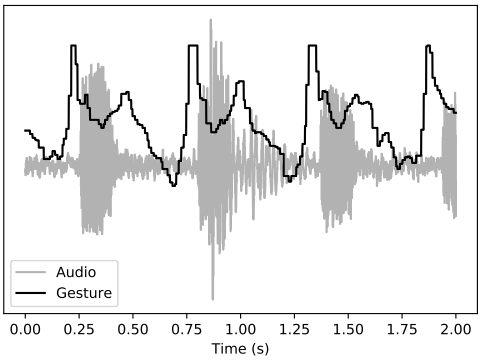


Figure 7: Audio and gesture related to a steady rhythm performed by S6.

between the motion signal valleys and peaks and vocalization data seems to be consistent for each subject. All these data suggest that different people imagine different interactions with the virtual instrument regardless of their previous musical experience.

3.2 Impact of Rhythm

When playing in a varying rhythm, subjects expressed less confidence during data acquisition. We speculate that this is linked to the fact that most people are more used to playing and listening to music with a steady rhythm. Moreover, performing an unknown rhythm in an unknown instrument generated discomfort.

Figures 8, 9, 10, 11, 12 and 13 show data acquired from the same subjects discussed in Section 3.1.

The subject in Figure 8 did not show consistency in the alignment of voice and gesture. This behavior replicates the observation in Figure 2. Moreover, the experienced musicians, as shown in figures 9, 10, 11, 12 and 13, stopped presenting alignment consistency.

Therefore, in all cases is possible to observe inconsistency in the alignment between gesture and vocalization. Also, the shape of their gesture signal varied more within the same acquisition.

Data suggests that subjects did not have a clear

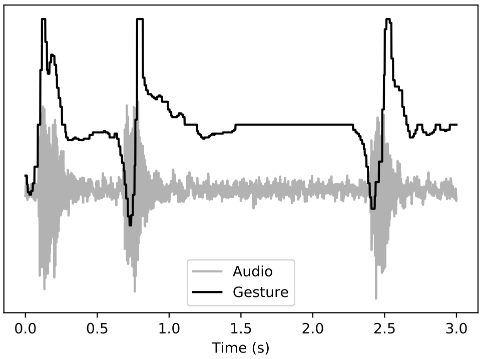


Figure 8: Audio and gesture related to a variable rhythm performed by S1.

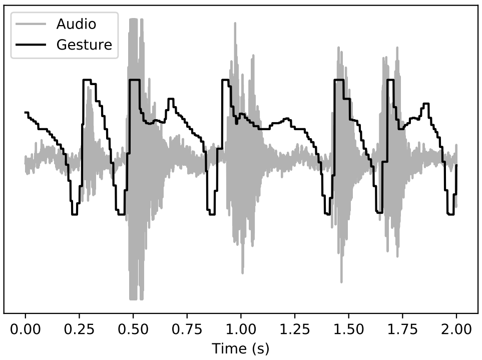


Figure 9: Audio and gesture related to a variable rhythm performed by S2.

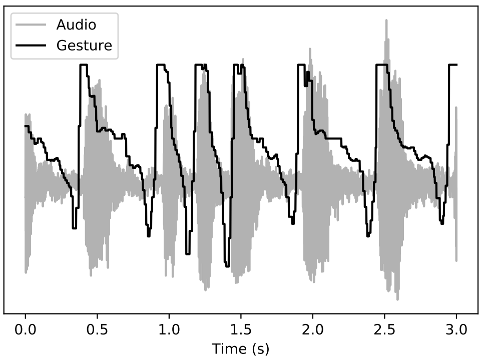


Figure 10: Audio and gesture related to a variable rhythm performed by S3.

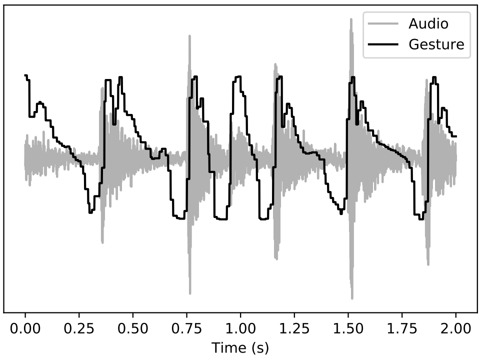


Figure 11: Audio and gesture related to a variable rhythm performed by S4.

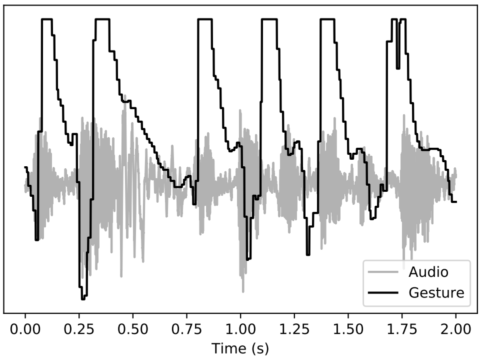


Figure 12: Audio and gesture related to a variable rhythm performed by S5.

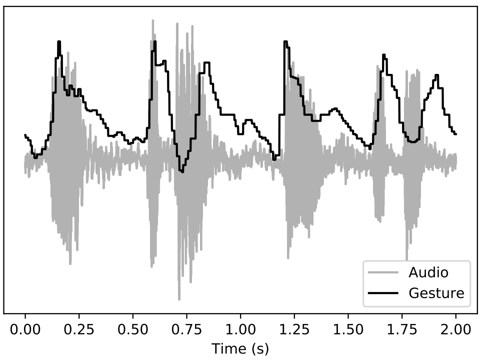


Figure 13: Audio and gesture related to a variable rhythm performed by S6.

idea of how to interact with the virtual instrument without the support of a predefined rhythm. This can be linked to their lack of experience with this specific instrument (imaginary vocalized drums) and these specific conditions, regardless of their general experience with music.

4 Discussion

There are two important aspects that must be noted in the acquired air drum gestures. First, we note that subsequent gestures performed by the same user tend to be similar. Second, we note that gestures performed by different users tend to be different.

The alignment analysis can be performed using the vocalization and the motion signal peaks as references. It is possible to see that S6 performs motions that peak around 0.1 s before the vocalization peak, as shown in Figure 7, while S5 (Figure 5) aligns motion and vocalization peaks and S1 (Figure 2) performs the peak around 0.2 s after the vocalization. This means that the peak can have a difference of up to 300 ms in the alignment due to the change in the subject. Such a difference is harmful for drum performances.

Similarly, the valley in the motion signal can happen together with the vocalization (S1), around 0.1 s before it (S2), immediately before the vocalization (S5) or up to 0.2 s before the vocalization (S6). This is means that this inter-subject difference is around 200 ms, which is also harmful for drum performances.

It is important to remember that real drums provide both audio and physical feedback. Moreover, the physical feedback strongly correlates to the audio feedback, both in their time alignment and their percussive, "point" quality. As a consequence, it is possible to learn and adapt oneself to the playing of a drum.

On the contrary, air drums are played solely using muscle memory and one's perspective. Even if virtual drums can yield audio feedback, they cannot provide physical interactions. Hence, the playing differences are hard to overcome by practicing.

For this reason, user-specific personalization is an important feature for virtual, accelerometer-controlled drums. This is a seldom explored problem in the field of digital musical instruments. In order to foster this type of research, we made our dataset available online at `http://timba.nics.unicamp.br/mir_datasets/gesture/wiimote_ajpc.zip`.

5 Conclusion

In this work we built a dataset containing both gestures and vocalizations related to a virtual percussion instrument imagined by subjects. This dataset is available at `http://timba.nics.unicamp.br/mir_datasets/gesture/wiimote_ajpc.zip`. We analyzed the shapes of the motion signals and their alignment to the corresponding vocalizations.

Qualitative analysis revealed that different persons use diverse motions to play imaginary drums, which corroborates with the observations of Maki-Patola [5]. Also, we see a high inter-subject difference between the alignment of peaks and valleys of the motion signal to their vocalizations is severely different. Nevertheless, we can see a greater intra-subject similarity between gestures when playing steady rhythms, but this similarity decreases when non-steady rhythms are played.

This means that predicting the vocalization beats is a task that requires user-specific personalization. Such a task wil be tackled in future work.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior - Brasil (CAPES) - Finance Code 001.

References

[1] K. Okada, F. Ishizawa, A. Kobayashi, A. Yoshii, M. Sakamoto, and T. Nakajima. Virtual drum: Ubiquitous and playful drum playing. In *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on*, pages 419–421, Oct 2014.

[2] Mike Collicutt, Carmine Casciato, and Marcelo M. Wanderley. From real to virtual : A comparison of input devices for percussion tasks. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 1–6, Pittsburgh, PA, United States, 2009.

[3] Thomas Hermann Tobias Grosshauser, Ulf Grossekathofer. New sensors and pattern recognition techniques for string instruments. In *New Interfaces for Musical Expression*, pages 271–276.

[4] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Ed).* Wiley, 2001.

[5] Teemu Maki-patola. User interface comparison for virtual drums. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 144–147, Vancouver, BC, Canada, 2005.

[6] Christophe Havel and Myriam Desainte-Catherine. Modeling an air percussion for composition and performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 31–34, Hamamatsu, Japan, 2004.

[7] Luke Dahl. Studying the timing of discrete musical air gestures. *Comput. Music J.*, 39(2):47–66, June 2015.

Sustainable Interfaces for Music Expression

Igino de Oliveira Silva Junior[1], **Flávio Luiz Schiavoni**[1]

[1] Arts, Urbanities and Sustainability
Arts Lab in Interfaces, Computers, and Everything Else - ALICE
Federal University of São João del-Rei
São João del-Rei, MG

igino@ufsj.edu.br, fls@ufsj.edu.br

Abstract. *The creation of Digital Musical Instruments (DMI) tries to keep abreast the technological progress and sometimes it does not worry about some possible side effects of its development. Obsolescence and residues, rampant consumption, constant need to generate innovation, code ephemerality, culture shock, social apartheid, are some possible traps that an equivocated DMI development can bring up to society. Faced all these possibilities, we are trying to understand what can be a sustainable Digital Instrument analyzing several dimensions of sustainability, from economical to cultural, from social to environmental. In this paper, we point out some possibilities to try to reach up more sustainable instruments development bringing up the human being and values like cooperation and collaboration to the center of the DMI development discussion. Through some questions, we seek to instigate a paradigm shift in art-science and provide a fertile field for future research.*

1 Introduction

Recently there has been a multitude of discussions about technological developments involving Digital Musical Instruments (DMI). This work intends to discuss the future of these instruments, especially in their social and cultural influences in order to promote a more sustainable relationship between people and these innovative products. By that, it is expected to think more sustainable DMIs, knowing how to meet people's current needs and allowing future generations to meet their own needs under an environmental, economic, social and cultural bias.

The creation of digital instruments tries to keep up with technological progress and often does not worry about the waste created by it, with the obsolescence of these interfaces and the rampant consumption that can be generated from the need for novelties tied to technological development. This points us to a first dimension of sustainability, **Environmental sustainability**, which can be thought of in the creation of these instruments. Of course, in bringing the issue of consumption, we also raise the possibility of discussing the **Economic sustainability** of these new instruments.

Some authors have already dealt with this factor in their research. Adrian Freed proposed a rework of the Stylophone integrating sustainability, longevity and design of instruments what generated the Fingerphone (Figure 1). This DMI "achieves low total material use, low energy cost and a small carbon footprint by using comparatively thin materials, recycled cellulose and carbon to implement the functions of the Stylophone without its high-energy cost and toxic materials: plastics, metals, glass fiber and resins"[1].

In addition, technological development and consumerism of new technologies can bring another imbalance up from a sustainable point of view, once this development disregards part of the population. **Social sustainability**, which proposes to bring social justice, democracy and peace culture, can be shaken if not thought of in this development. According to Barbosa et al. [2] "the design process is not focused on creating DMIs for a community with a particular culture - with established instruments, repertoire and playing styles - outside European and North American traditions". At many times, the builders of new musical interfaces advocate that new music is needed for these interfaces to be used. Creating a new music is not exactly a problem, but integrating these instruments into popular and already existing music can instigate the concept of **Cultural sustainability**, another aspect that must be brought to the light.

In this article, we will bring some discussions about sustainability and new interfaces taking into account these dimensions mainly in aspects of collaboration, interaction, creation, transparency and empathy with contemporary society. We aim to provide a breeding ground for theoretical, technical and artistic development for future studies in this area.

2 The dimensions of sustainability

Nowadays there are many discussions about what sustainability is and all its dimensions. A very general definition was set out in the UN report "Our Common Future" by Gro Brundtland in 1987: "Humanity has the ability to make sustainable development to ensure that it meets the needs of the present without compromising the ability of future generations to meet their own needs" [3]. Although this concept is correct, some authors like Leonardo Boff affirm that it is problematic when considering only an anthropocentric aspect of sustainability, not addressing an environmental aspect that is consistent with all forms of life [4]. Obviously discussing the dimensions of sustainability is seeking to integrate all its demands, be it environmental, economic, cultural or social demand.

Therefore, sustainability will be treated here as any action designed to maintain the social-political, energetic, informative and physical-chemical conditions that sustain all beings, especially the community of all types of life, human life and its cultural expressions, aiming at its continuity and to meet the needs of the present and future generations [4]. Our greatest doubt is how to meet these demands from the perspective of digital art and technology since both are intrinsically linked to contemporary everyday life.

In this way, thinking about sustainability through digital art means thinking about a computer that embraces education, communication and creative processes, even if it is about digital musical instruments, so that this environment is open and democratic for the whole population. Much of the widespread poverty, environmental desecration, and waste of human life seen around the globe could be prevented by known (to humanity as a whole) technologies, many of which are simply not available to those that need it [5].

Also, thinking about sustainability in this kind of art permeates the dialogue and the active participation of the public. Based on the concepts of interactive art raised by Julio Plaza, the beginning of dialogism appears in the language studies of Mikhail Bakhtin, since "every sign results from a consensus between socially organized individuals in the course of a process of interaction (...) which must not be dissociated from its material reality, from the concrete forms of social communication" [6]. So, with the playful participation and the creativity of the spectator, the concepts of "art for all" and "do it yourself" fit in all forms of art, including the creation of DMIs.

2.1 Environmental impact

Since the industrialization process took part on the humankind history, enormous quantities of garbage have been thrown into the environment, heavily influenced by a society based on rampant consumption. This environmental impact can be reduced by adopting attitudes of reuse, not only of materials like hardware, but also of codes, music and art pieces. If hardware reuse is obvious a good thing to the environmental sustainability due the save of primary matter, code reuse can save energy, a resource that normally is produced by environmental damage.

In this regard, the environmental agenda can also be a theme to be explored by art, taking into account the current nature of our planet. A diseased planet, but through art we have the ability to help other people to understand their reality, creating a new mental paradigm in which we understand to be really part of nature and its cycles. The transition to a sustainable society must be guided by basic principles such as the reduction of consumption; the reuse and recycling of hardware and software. The question is: What would be a musical interface that causes less environmental impact?

2.2 Economic impact

In the case of economic sustainability we are faced with a dichotomy between the current economic system and the natural life system. The current economic system seeks to increase production in favor of consumption, which intrinsically requires the depredation of part of the planet and, as

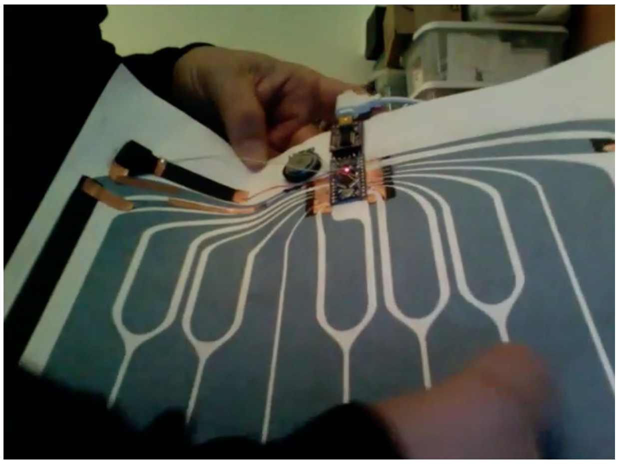


Figure 1: According to Adrian Freed, the first Fingerphone version was created to demonstrate the conductivity of the paper in East Bay Maker Faire. The next version used a pizza box as a resonator to emphasize reuse as the sustainability aspect.

a consequence, the production of perverse social inequalities. In contrast, the natural life system aims at balancing all the factors so that the planet can maintain its capacity to replenish the resources used by us and the integrity of its nature [4].

Figure 2: "Sobe Aí" was a Jam Session held in 2014 and has presented the last evaluation phase of the instruments created by the Batebit group. During this event, all those who worked in the "Edifício Pernambuco" sought to expose to the public what they were working into.

During the process of creation of new technologies or new art pieces, our intention is to think more about how I can produce in balance with nature, than how much I can profit. It is fact that technology is money, Art is money, and DMIs are technologies and Art, but how can the negative economic impact of a new instrument be mitigated? Investment in tech companies worth a lot, the investment in art does it too but what are we using these investments for?

For example, digital music, through MP3, has changed the world economy by taking away the economic power of record labels and taking them to online businesses. Does it change something for those who wants to listen music? We are moving from traditional instrument development to digital instrument development. Does it change something for those who wants to play? Is it at least more accessible?

Often, as researchers, we think of having a profit to pay for our work and this is important as well, but how not to have a negative impact? Is it possible to create richness without generating poverty? Can we do it with art and our research? If we do not consider this situation, the degradation will continue to a no longer sustainable limit that is being revealed by global warming.

One of the paths to be followed is the establishment of networks of reciprocal exchanges of knowledge, also known as Solidarity Economy. Some examples of organizations are Fair Trande, Co-operatives, Trade Unions, Social Centers, Give-away shops and Free Software movements. The structure of these networks allows non-hierarchical relationships between suppliers and users of knowledge of all kind increasing the quality of life of a region. The formation of human and material resources for Solidarity Economy projects, especially during music festivals (i.e. digital Jam Sessions), which have the greatest potential for generating work and income, can contribute to a higher productivity in the music sector.

2.3 Cultural impact

Unfortunately, for many years, the cultural and social dimension of Sustainability played only a tangent role in the debate and to the focus of this area of study. Only in recent years has these dimension become more conscious, as we can note in the UNESCO Action Plan "The Power of Culture" (Stockholm, 1998) and UN-Decade 2005-2014 "Education for Sustainable Development". These studies are important because they make clear that if part of the global crisis has cultural causes, then we need to think about cultural solutions [7].

Figure 3: Evaluation of Giromin and Pandiva, instruments created by the Batebit group, with Jam da Silva.

The transformation of local society by the dynamics of a global society, specifically in the creation of digital instruments, allows a new reflection on the impacts of music technology on traditional cultures, especially on the trajectory of a media society and constantly submitted to the consumption of symbolic goods. In order to adapt to a more sustainable global social-political development agenda, we need to adopt new cultural sustainability strategies in harmony with a local agenda.

According to Davide Brocchi, cultures do not just define the boundary between the social system and the environment: they control communication and exchange between these systems, therefore cultures define groups of societies and subcultures - and vice versa [7]. To be in tune with cultural and social sustainability is to seek to understand theoretically the complexity of contemporaneity, the uniqueness of the composite character of its sociability and the meaningful inscription of communication, especially in its mediated version, in this peculiar structured circumstance acclimated by the media.

We must bear in mind that it is the place that provides an important part of the sense of belonging and identity, as well as being partly responsible for the generation of cultural expressions. In this regard, Luke Seghezzo quotes McShane in stating that feelings and moral lives are lived from the inside, in the first person [8]. Therefore, we should not only concern ourselves with the material products of DMI, but also with the inner life of the being that produces and consumes these products. To do so, we do not just need an education for sustainability, but probably also different media and communication structures [7].

The project developed in the Brazilian northeast called Giromin [2] (Figure 3) chose to apply a different modus operandi in the development of DMI. Through the methodology of Design Thinking, the group established ongoing contact with local musicians and developed their DMIs under what they called a more holistic perspective. "Musicians see their instruments as an unit, which means that timbre, gesture control interface, ergonomics and appearance are inseparable. This continuous contact allowed us to distance ourselves from a more technical perspective (i.e., what sensors and input technologies should we use?), and to get closer to the reality of musicians" [2].

From the point of view of cultural sustainability, this form of approach has the potential to guarantee self-confidence and self-determination of the identity of a population. According to the authors of the Batebit project, their greatest learning was that "DMIs should be somehow related to the instruments (and their respective playing styles) used inside the community; DMIs should allow musicians to perform the community's standard repertoire and DMIs should allow musicians to use the community's standard gestures and accessories" [2].

2.4 Social impact

Contemporaneity, apprehended as a society based on communication, has transformed information into one of the most valuable commodities. On the one hand, information runs from the means of production to consumption, giving support to a society of consumption and symbolic goods. Thus, information has the capacity to manipulate population activities in a social-political scope, depending on the purpose with which it is being used. Following this line of thought, we are faced with the unsustainable attitude of exercising on a local musical knowledge a global and hegemonic musical knowledge for a specific group, directly influencing the culture of a given people through a hierarchical relation of knowledge.

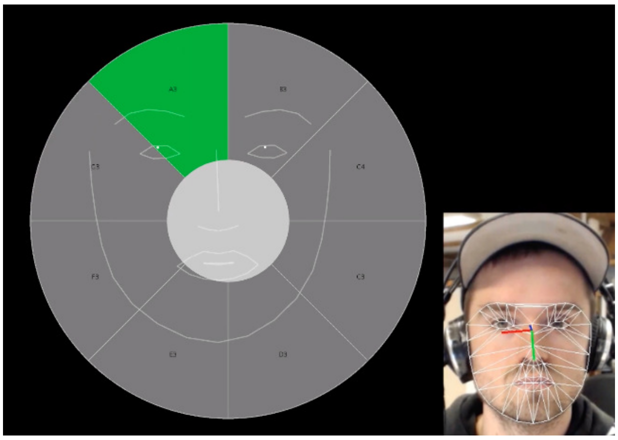


Figure 4: Eye Conductor is a customizable musical interface, created in an academic project by Andreas Refsgaard, with facial recognition that helps people with physical disabilities create music through eye movements and facial expressions.

In the dynamics of our society we find groups that are dominant and groups that are dominated which results in processes of exclusion that are often inhuman. When we consider social inclusion in our activities we are defining a range of activities that ensure the democratic participation of all in society, regardless of religion, gender, sexual orientation, education, social class or physical condition.

We often generate products thinking exclusively for a hegemonic group of people, but how can digital musical instruments contribute to the inclusion, participation, and socialization of people with cognitive, motor, and social limitations? How many people can be left out of the creation and use of these instruments? How does this not end up creating a barrier between the included and the excluded? By promoting free access to DMI programming, we can use it to raise awareness, teach programming, digital inclusion, and technology dispersion.

Finally, after pondering these issues, we must keep in mind that technology is a fundamental aspect of human existence in the contemporary world, and therefore, it is important for our society to include the production of DMIs in a global agenda for sustainable development. All aspects should be taken into account. We urgently need to think about recycling, adapting to existing cultures and not generating extremely expensive products to allow the population's accessibility. One of the possibilities to make this viable lies in technological openness and coding reuse.

3 The openness as a possible impact reduction

The concept of technological openness covers a broad range of things. Somehow, the concept of technological openness is on the opposite side of proprietary technology. Normally, people does not think how copyright, patents, and industrial secrets influences how we deal with our technological devices. In fact, there are a lot of technology that is not open because it belongs to a person or a company and one is not allowed to use it without the right permission. The idea of technological openness is a technology that can be used freely, and "free" here is not like "free beers" but like "free speech" [9].

It is possible to define our tech devices as a stack. At the top level we have data, applications, user interfaces. In the medium level we have files, operating systems and network protocols. At the bottom, the hardware. All these levels can have or should have an open choice of technology.

How the open technological choice can impact Digital Arts and sustainability? We chose some border interest field of art where the impact is clear.

3.1 Economics

A first impact of adopting open technology in art projects is on budget. Here, the "free" concept means with no costs. Therefore, a project based on FLOSS can have a reduced cost or no cost at all with software and licenses if compared with proprietary tools. Whereupon some people claims that open technology is harder to learn.

Assuming that open tools can be less intuitive than commercial solutions, what certainly is not true, the money saved with software license can be invested on learning an open technology if an abrupt learning curve really exists. The difference between both investments is that a software soon or later become obsolete but knowledge lasts a long time. Besides, the investment in learning can be local while the investment in buying uses to be global. If the investment considers to create a new software, a FLOSS existing application can be a start line to create the new solution. All in all, open source is more efficient and adaptable than proprietary closed, hierarchical systems [5].

3.2 Collaboration

According to Edmonds et. al. [10], collaboration in art practice has grown significantly since art adopted digital mediums to share data. Digital art gave a next step on collaborative art and the collaboration became huger evolving artists and other people from different disciplines with different skills [11]. Computers are the perfect machine for perfect copies and digital is the perfect medium to share data.

Since collaboration in digital art is possible and easy, what can disturb it to happen? The answer is obvious: Non shareable content. It is not possible to collaborate if we are dealing with copyright, patents and secrets.

The choice for open tech is evident when there are several people working together to develop a piece of art, like an instrument or a composition.

It can be necessary to use several different tools to create an art piece and it will be easier if they choose open file formats to exchange data between all these tools. A choice for FLOSS can easy the share of tools and applications, no piracy, no high costs to invite collaborators. The usage of open protocols can also easy the sharing between an art team.

3.3 Creativity

Creativity in arts is an obscure but fundamental process that involves imagination, exploration, discovery, experimentation and curiosity. In digital arts, it involves more than simply learning a new software. A software brings a lot of concepts in its design, concepts that presents the creativity of the software developer. Experiment with the software can lead to learn how the software was developed and how it can be hacked to a new functionality [12], adding new features that presents new artistic concepts.

Thus, A digital artist can consider computer technology as a way to enhance creativity [13], specially when it is possible to explore the software without the borders imposed by the developer. Develop a software is a work of creativity and can be an artistic work, based on experimentation evolving aesthetics meanings and not only functional questions [11].

Regarding creativity, open source software, open hardware, open data, protocols and file types can be a fertile kingdom to digital artists. Open technologies allow an artist to modify, correct, integrate, cut, copy and paste existing technologies to create new artifacts according to their personal needs [13].

3.4 Education

In education, open technologies works on the border of art and several other fields like Engineering, Math, Computer Science, Statistics, Physics and Telecommunication. It is almost impossible to learn digital art without understanding how computers work and the main idea of understand how a software works is the base of the FLOSS movement.

For this reason, Open technology in arts can be a great phenomenon for trans-disciplinary dialog [14]. An Art specialist needs the learning experience of several different areas, a strong educational approach not only for artists but for students which creativity and curiosity is an important value [12].

Another educational impact regards the relationship between students and technology. Computers tend to be a genius machine if one does not understand how it works. Using open technology is possible to demystify the computer and learn how it works and how it can be changed. It has good consequences in social and digital inclusion.

3.5 Adaptation

Open software can be adapted for other purposes. This action can facilitate its inclusion in other social environments taking place in diverse cultural and social inclusion projects. Even if software adaptation is one of the ways to improve productivity and quality of software, it is known that producing software with adaptability and high quality is still a complex process. In order to achieve the sustainability that we seek, we need to institute a culture of reuse in the development process, remembering that this can give us a more democratic environment by reducing the effort development and maintenance of a completely new system. For programmers and artists with a local agenda this can lead to reduced development time, reduction of error density, reduction of maintenance costs and reduction of total development costs.

4 Final Considerations

The purpose of this article was to raise some reflections about sustainability and its influence on technology and art, understanding this relationship as a viable alternative to achieve a certain degree of freedom and democracy in the creation of DMI, since expanding access to technology is an aspect of necessity human basis.

In addressing the dimensions of sustainability from the environmental to the social, we argue that technology must be distributed and shared not only as a product, but as a code that is part of a new mental paradigm that can be adapted to each local agenda according to the culture of a population and social inclusion.

Open technology has the potential to mitigate computational tools that create exclusion of people, either through accessibility based entirely on consumption or hardware and software waste. We are proposing to avoid this segregation and to consider in a technological adhesion that is sustainable and free. Obsolescence and waste, rampant consumption, constant need to generate innovation and code ephemerality were some of the problems we discussed and seek to minimize by proposing access to the source code of the products generated.

In order to provide greater knowledge to future researchers in this area, it is expected that the points discussed will instigate more dialogic relationships in our society by changing their rhythms, aspiring to a more powerful use of digital art and full enjoyment of their rights. Thus, local artists are not dependent on developers and can play a collaborative role in producing art, music, and interfaces for musical expression.

Once, someone told us that if you made a question in a scientific paper, you should answer it in the same paper. We are sorry if we did not answer all the questions we made. We do not have the answers and, believe us, we would love to have. We are worried about it and we know other people are worried too. Art is about creativity. We are using our creativity to think about these issues and we hope more people join us on it.

References

[1] Adrian Freed. The fingerphone: a case study of sustainable instrument redesign. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan.

[2] Jerônimo Barbosa, Filipe Calegario, Jo ao Tragtenberg, Giordano Cabral, Geber Ramalho, and Marcelo M. Wanderley. Designing dmis for popular music in the Brazilian northeast: Lessons learned. In Edgar Berdahl and Jesse Allison, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 277–280, Baton Rouge, Louisiana, USA, May 2015. Louisiana State University.

[3] World Commission and Development on Environment. Our common future. sog. brundlandt-bericht, United Nations, 1987.

[4] Leonardo Boff. *Sustentabilidade: o que é - o que não é*. Editora Vozes, 2017.

[5] Joshua M. Pearce. The case for open source appropriate technology. *Environment, Development and Sustainability*, 14(3):425–431, 2012.

[6] Julio Plaza. Arte e interatividade: autor-obra-recepção. *ARS (São Paulo)*, 1(2):09–29, dez. 2003.

[7] Davide Brocchi. The cultural dimension of sustainability. *Religion and Dangerous Environmental Change: Transdisciplinary Perspectives on the Ethics of Climate and Sustainability*, 145, 2010.

[8] Lucas Seghezzo. The five dimensions of sustainability. *Environmental politics*, 18(4):539–556, 2009.

[9] Richard Stallman. *Free software, free society: Selected essays of Richard M. Stallman*. Lulu. com, 2002.

[10] Ernest Edmonds, Greg Turner, and Linda Candy. Approaches to interactive art systems. In *Proceedings of the 2Nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, GRAPHITE '04, pages 113–117, New York, NY, USA, 2004. ACM.

[11] Ernest Edmonds and Linda Candy. Creativity, art practice, and knowledge. *Commun. ACM*, 45(10):91–95, October 2002.

[12] Joanna Black and Kathy Browning. Creativity in digital art education teaching practices. *Art Education*, 64(5):19–34, 2011.

[13] Adérito Marcos. Digital art: When artistic and cultural muse merges with computer technology. *IEEE Computer Graphics and Applications*, 27(5):98–103, Sept 2007.

[14] Georg von Krogh and Sebastian Spaeth. The open source software phenomenon: Characteristics that promote research. *J. Strateg. Inf. Syst.*, 16(3):236–253, September 2007.

Ha Dou Ken Music: Mapping a joysticks as a musical controller

Gabriel Lopes Rocha , João Teixeira Araújo , Flávio Luiz Schiavoni

[1] Arts Lab in Interfaces, Computers, and Everything Else - ALICE
Federal University of São João del-Rei - UFSJ
São João del-Rei - MG

gbr.cdc@gmail.com, teixeira.araujo@gmail.com, fls@ufsj.edu.br

Abstract. *The structure of a digital musical instrument (DMI) can be splitted up in three parts: interface, mapping and synthesizer. For DMI's, in which sound synthesis is done via software, the interaction interface serves to capture the performer's gestures, which can be mapped under various techniques to different sounds. In this work, we bring videogame controls as an interface for musical interaction. Due to its great presence in popular culture and its ease of access, even people who are not in the habit of playing electronic games possibly interacted with this kind of interface once in a lifetime. Thus, gestures like pressing a sequence of buttons, pressing them simultaneously or sliding your fingers through the control can be mapped for musical creation. This work aims the elaboration of a strategy in which several gestures captured by the interface can influence one or several parameters of the sound synthesis, making a mapping denominated many to many. Buttons combinations used to perform game actions that are common in fighting games, like Street Fighter, were mapped to the synthesizer to create a music. Experiments show that this mapping is capable of influencing the musical expression of a DMI making it closer to an acoustic instrument.*

1 Introduction

The current state of the technology allows relatively inexpensive access to Personal Computers with some reasonable processing capacity. This facilitates the creation of live musical performances through the control of sound synthesis by computers in real time. Researches in the field of Computer Music led to the development of several models of sound synthesis for instrument creation. Such models can be classified between physical models and signal models [1].

Those of the first category are based on the physical and acoustic events related to the sound production of a given instrument [2]. Through the analysis of these instruments it is possible to develop a system of equations to simulate this instrument in a realistic way. Signal models, on the other hand, rely on the spectral representation of the sound and signal processing structures to produce the desired sound [3]. Such models enable the creation of sounds that do not resemble those of existing acoustic instruments.

However, the standard input interfaces used in desktop or laptop computers can be not adequate when searching for expressive control of the synthesis parameters of a digital instrument. Such interfaces were developed from an ergonomic approach and are focused in traditional tasks like typing or drawing. Their goal is to be simple, easy to learn and objective. On the other hand, the role of creativity has great importance in the development of controllers for musical interaction [1]. These controllers are more expressive but also are projected to be used by an expert skilled performer. The learning process for using these interfaces can require a certain amount of effort that can be prohibitive for lay musicians interested into explore digital instruments. This is an important characteristic of a digital musical instrument because it is possible to make a correlation between effort and expressiveness[4].

Beyond the interface and the synthesizer, a digital musical instrument also has a kind of hidden layer also known as mapping. For digital musical instruments, mapping[5, 6, 7] consists of the task of making the connection between control parameters, related to the sensors present at the interface, and synthesizer parameters, related to the chosen sound synthesis method. There are several possible approaches to mapping. Commonly it is designed in a simplistic way, mostly with one-to-one connections between parameters. This practice is contraindicated because it tends to generate an instrument with poor expressive abilities. If we think the expressiveness of an acoustic instruments, since they are governed by physical laws, control methods are often interconnected in a complex way with sound results. A researcher might take this into account in the mapping process when developing a new digital instrument, specially if the goal is not to copy an existing acoustic instrument. The chosen strategy has great impact in achieving an expressiveness close to that of an acoustic instruments.

In our work, we did not aimed to create a new musical interface, but we reused one developed for another context. We used a video game controller, specifically the DualAnalog model originally developed by Sony company for its Playstation console, as a musical interface. This choice had three main reasons. The first is the ease of access at low prices, whether in specialty stores, the Internet or popular markets. The second is the state of ubiquity given to this technology by the strong presence of video games and electronic games in popular culture. Most people already have some experience with this interface or similar, even if not strongly attached into video games. The third is an emotional and aesthetic question. This is an interface that brings to many gamers a lot of good memories. In the authors' experience in using this interface in performance, people tend to be impressed when they see a video game controller being used for musical purposes.

Our interface can be classified as alternate controller, since it is not based on any existing acoustic instrument. This makes the task of mapping more difficult because the possible relationships between control and synthesis parameters are numerous and not obvious. To overcome this difficulty, in this work we suggest to analyze how these interactions are performed in electronic games, besides using them as inspiration for our mapping. In our DMI we have selected the case of arcade-style fighting games such as “Street Fighter” and “The King of Fighters” and the way special attacks are performed on them.

2 Related Works

Matthew Blessing and Edgar Berdahl [8] developed, at Louisiana State University, a control similar to the one presented in this work. Authors used as input interface an Arduino Micro with five levers, where the X and Y position of each lever determines the changes in range and pitch of the sound. As a synthesizer, this work uses a Raspberry Pi running a PureData patch. In addition, the instrument was mounted in an enclosure with input interface, sound synthesizer, mapping and sound box.

In the music department of Stanford University, SookYoung Won, Humane Chan and Jeremy Liu [9] developed an instrument consisting of aligned PVC tubes containing photovoltaic resistors at the bottom of each tube. This project uses a microprocessor as a data processing unit, which transforms the signal sent by the photovoltaic resistors, when triggered by the incidence of light, into a MIDI signal. However, the data coming from the photovoltaic resistors are only considered as notes, varying initially, from note 60 to 72 and then maintaining the MIDI velocity value at 64 as the default for all notes. The project also has the possibility of varying scale, which they call “high” and “low”, being able to play MIDI notes from 48 to 60, and from 72 to 84. After the signal is transformed into a MIDI signal it is sent to the PureData patch, which contains the sound synthesis, and then the sound is generated.

Christopher Ariza [10] also explores the DualAnalog model as a control possibility for musical interaction. He discusses the characteristics of this control, its possibilities and performance opportunities. Practical uses for the button set and mappings applied to the pair of joysticks are presented, such as the use of continuous axes to send MIDI messages and to control the cutoff frequency of a low-pass filter, which are also used in our work. For Ariza, the pair of joystcks carries the most expressive possibility in this type of control. He presented in his work a set of detailed mappings that implement 11 different instruments divided into 6 categories, each with its characteristics and applications. Such implementations are done using a Pd library called Martingale Pd, but the possibility of using other software is discussed. It even features a way to quickly switch between different presets. In this way, the artist will have access to a wide range of instruments for his performance.

Several papers explore the possibility to use the control used on the Nintendo Wii, equipped with motion sensors that allow the capture of more complex gestures, as musical interface [11] [12].

3 The joystick as a DMI

To present the construction of our musical interface, we divided our instrument in three parts and we are presenting it part by part: The interface, the synthesizer and the mapping strategies. Certainly, the presented project reflects several development decisions that can be totally different in a similar project.

3.1 The Interface

Among the various video game controls currently on the market, the popular model of the Sony’s PlayStation, called DualAnalog (later Dualshock). Many variants of its first version were launched by various companies like the Double Shock B-Max Controller, which includes a Universal Serial Bus (USB) connection to be connected to a Personal Computer (PC), used in this work. This type of control can be found easily at low prices and offers a number of attractions if compared with similar interfaces. It is practically impossible to achieve this amount of buttons and analogue levers available for mapping at such a low cost if the option were to build such a circuit electronically.

About the control, it is worth mentioning that it can be used in two modes, with analog mode on or off, selected with the “Analog button”. When the analog mode is disabled, the continuous axes assigned to the levers are not used. Thus, the left joystick controls the discrete axis of the directional buttons and the right joystick can be used to trigger the right-hand buttons. This configuration is interesting because it allows two ways to execute the same command, even if the number of available commands is somewhat limited.

As can be seen in Figure 1, this control has

- (1 to 4) four front buttons (right hand - popularly known as “triangle”, “x”, “square” and “circle”);
- (5 to 8) four upper buttons (“L1”, “L2”, “R1” and “R2”);
- (9 to 12) four directional buttons (left hand - also known as D-pad);
- (13) a start button;
- (14) a select button;
- (15 and 16) two analog joysticks;
- (17 and 18) two buttons pressing the levers;
- (19) an analog mode (on/off) button.

Currently, there are APIs in the Linux operating system that allow interaction with events triggered by video game controls. There is even a Webaudio library that runs this function in a web browser, the gamepad API. In this work, specifically regarding the implementation, the Linux library “joystick.h” was used to interact with this device.

The joystick.h library, available for Linux, allows the programmer to interact with the controller interface. It has a structure that encapsulates the information, whether

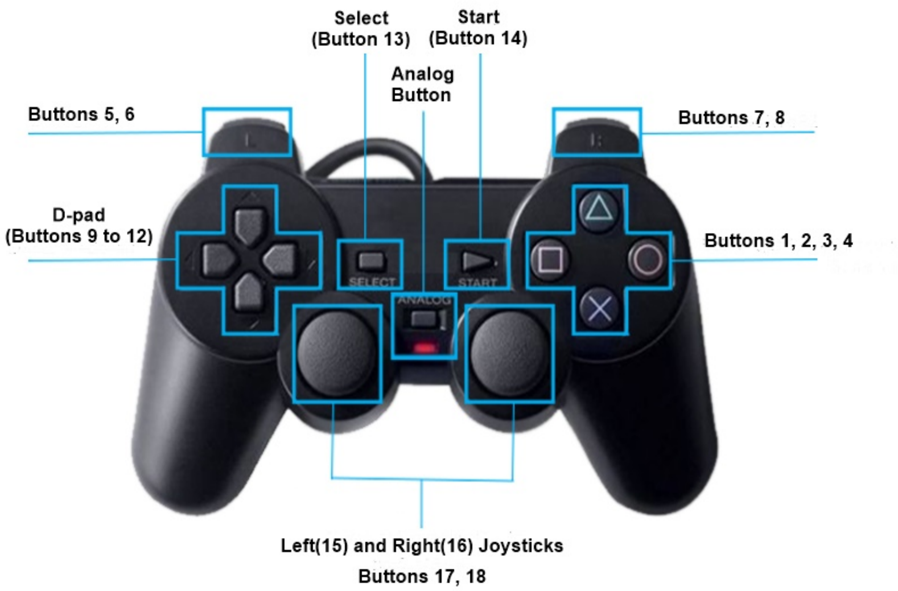


Figure 1: Joystick controls used in this work

Table 1: Control outputs and possible values

Buttons	Value
1 to 4 (front)	0 / 1
5 to 8 (upper)	0 / 1
13 - Start	0 / 1
14 - Select	0 / 1
15 - Left Analog	0 / 1
16 - Right Analog	0 / 1
19 - Analog	-
Axes	Value
9 to 10 - D-pad X	-32767 / 0 / +32767
11 to 12 - D-pad Y	-32767 / 0 / +32767
15 - Left Analog X	-32767 to +32767
15 - Left Analog Y	-32767 to +32767
16 - Right Analog X	-32767 to +32767
16 - Right Analog Y	-32767 to +32767

sent by the buttons or sent from the movement of the analog joysticks. In this way, the value of a button can be discrete, meaning only if it has been pressed, released or continuous, indicating the amount of movement of the joystick. It is noteworthy that the directional buttons, although discrete, control two axes, being one horizontal and one vertical, as well as the two joysticks. The events captured by this library is also timestamped and it is possible to capture the time in which each these events occurred. All the buttons and axes with their respective outputs are presented in Table 1.

3.2 The Synthesizer

Two options were considered when defining the synthesizer of our experiment: to use an existing synthesizers or to develop a new synthesizer in an sound programming environment like Pure Data. In this work, it was decided to use a MIDI protocol interface to connect with existing synthesizers, allowing to connect the a type of mapping in different synthesis engines, making the process of creating a instrument more free and interesting.

A MIDI synthesizer has a set of specific and predefined inputs as well as a well-defined communication protocol based on Notes and Dynamics (velocity). Although we find it sufficient for our initial proposal, we understand that this gives us a sparse set of synthesis parameters to work on mapping.

We have a wide variety of compatible synthesizers available for Linux, which makes it possible to explore several results with great ease. In this work, we used the set of synthesizers available in a tool called LMMS - The Linux Multimedia Studio. The LMMS, presented in Figure 2, is a multi-platform digital audio workstation that was widely used in our research.

Apart from a large number of interesting synths, the LMMS has a default MIDI interface to all synths, listed in the table 2, with several different parameters. Thus, it would be possible to use MIDI using control parameters, extrapolating the MIDI note events.

In these synthesizers it was possible to control the pitch, intensity and duration of the note. We can also work with LFO (Low Frequency Oscillator) and ADSR (Attack, Sustain, Decay, Release) envelope, which are common in several synthesizers. As for pitch control is possible to variate it over time in a way that make possible the execution of a glissando. With this, we maintain the advantage of being able to switch easily from synthesizer in order to explore new results.

To integrate the LMMS' synths in our project, we used the ALSA (Advanced Linux Sound Architecture)

Figure 2: Example of LMMS Synthesizer.

Table 2: Parameters accepted by the synthesizers

Parameter	Value
MIDI NOTE	
Note	0 to 127
Velocity	0 to 127
Envelope	
Attack	0 to 127
Duration	0 to 127
Sustain	0 to 127
Release	0 to 127
LFO	
Attack	0 to 127
Velocity	0 to 127
Filter	
Frequency	0 to 127
Resonance	0 to 127
General	
Volume	0 to 127
Pitch	0 to 127

MIDI API. Using this API, we were able to translate the events we captured from the control into MIDI protocol events. We build our prototypes in this way due to the great popularity and simplicity of handling of this protocol, which until today is widely used in the context of musical production.

3.3 1-to-1 mapping

Under the first formulated implementation, a simpler mapping was used to test the control possibilities. As mentioned previously, a 1-to-1 mapping is not indicated because it does not contribute to the search for expressiveness in the interface. However, this implementation served as a starting point for our research. Among those who tested it, the simple fact of turning a video game controller into a musical interface served as a great draw.

In this mapping we use the 8 main buttons, four front and four upper to trigger MIDI notes, organized in an octave, as musical notes. The D-pad serves to determine the speed parameter, where the value 80 corresponds to the neutral position and keeping it pressed under any of the directions can change it. The resulting values increase from the bottom to the top, from left to right. In addition, the left joystick can be used in the same way. The right lever is used to control the volume. Rotating it clockwise will increase it and counterclockwise will decrease it. The buttons pressed when pressing the joysticks serve to raise, with the left, and descend, with the right, an octave.

The start button serves as the Panic Button, it sends a NOTE_OFF for all possible notes to be used in case an error occurs and some note is lost. This implementation is recurrent in all mappings. The table 3 shows this mapping organization.

Table 3: 1-to-1 mapping with main buttons sending Notes and the D-Pad controlling Dynamics

Buttons	Mapped to
Button 1	NOTE 60
Button 2	NOTE 62
Button 3	NOTE 64
Button 4	NOTE 65
Button 5	NOTE 67
Button 6	NOTE 69
Button 7	NOTE 71
Button 8	NOTE 72
Left Analog	Octave down
Right Analog	Octave up
Start	Panic Button
Axis	Mapped to
D-pad Left	Velocity 50
D-pad Right	Velocity 100
D-pad Up	Velocity 127
D-pad Down	Velocity 30
D-pad Neutral	Velocity 80
Left Analog	Equals D-pad
Right Analog	Volume

3.4 1-to-many mapping

The 1-to-many mapping grants a macro-level control of the sound event, though it fails to give more detailed control over the parameters composing it [13]. In order to test such characteristic, we implemented the idea of using the D-pad or the left analogue not only to control the dynamics, but to change several characteristics of the resulting sound by pressing the buttons. This time, the use of diagonals was included to expand the possibilities. We started using the Envelope and LFO features present in the synthesizers using the MIDI control API. In the neutral position the sounds have short duration. As the left analog is moved clockwise, the sounds begin to acquire longer duration, lift and strength. The last three positions add the LFO oscillation. Finally, the rest of the control continues with the same functions as the previous mapping. This organization is shown on the table 4.

Table 4: 1-to-many mapping, with the right analogue speed controlling, Envelope and LFO

Buttons	Mapped to
Button 1	NOTE 60
Button 2	NOTE 62
Button 3	NOTE 64
Button 4	NOTE 65
Button 5	NOTE 67
Button 6	NOTE 69
Button 7	NOTE 71
Button 8	NOTE 72
Left Analog	Octave down
Right Analog	Octave up
Start	Panic Button
Axis	Mapped to
D-pad	Velocity, Envelope and LFO variation
Left Analog	Equals D-pad
Right Analog	Volume

3.5 Many-to-1 mapping

Advancing a bit more in mapping styles, the idea was to use some “many-to-1” mappings to extend control possibilities. For this purpose, instead of just using the 8 main buttons to trigger MIDI notes, it was also implemented the possibility of making combinations between them.

In total, 16 combination possibilities were created with the top four buttons, 4 with one button pressed, 6 with two buttons pressed, 4 with three buttons pressed, 1 with four buttons pressed and one with no button pressed. We use these buttons as a form of control in combination with others.

Thus, for each of the four buttons on the right side of the control, we have 16 note possibilities totaling 64 achievable notes. From the greater number of degrees of freedom achieved now, we have the possibility to also add functions to change the ADSR and the LFO.

Then, the right lever was used, which used to control only the volume. The characteristic and parameter are selected according to the combination of the buttons pressed. When the buttons are released, the vertical position of the lever determines the value of the parameter. Thus, our instrument ends up having different forms of configuration, although in this version of mapping, the control of individual parameters still leaves the task a bit complex.

For greater expressiveness, the left lever, which previously had the same effect as the directional ones, starts to control parameters such as the cutoff frequency and the resonance of a low pass filter, under its vertical and horizontal axes, respectively. The D-pad here continues to determine the velocity of the notes, however, to facilitate control, which has now become more complex, simply press one of the directional keys once to effect the change, in an on/off behavior. To reach the neutral position in this way, it is necessary to press the active position again. All the buttons and axes definitions are show on the table 5.

Table 5: Many-to-1 mapping, with the upper buttons selecting notes and parameters

Buttons	Mapped to
Button 1 to 4(frontal)	NOTE activation
Button 4 to 8(upper)	NOTE/Parameter selection
Left Analog	Octave down
Right Analog	Octave up
Start	Panic Button
Axis	Mapped to
D-pad	Velocity control
Left Analog X	Filter frequency
Left Analog Y	Filter resonance
Right Analog Y	Parameter value
Right Analog(Rotation)	Volume

3.6 Mapping many to many

In our final mapping, we have sought to take inspiration on commands used in fighting games such as Street Fighter. We thus seek to exploit greater expressive capabilities of video game control with the capture of complex movements. With this we can also explore the affective memory of people whose games were part of their childhood and adolescence.

Here the controls become somewhat more simplified than in the previous mapping because we abandoned the idea of using the upper buttons to select notes and parameters, transferring this function to the directional buttons. When making combos by sliding your fingers through the D-pad, in the classic style of Arcade, a number of parameters can be modified in the sound event to be executed, be it in pitch, intensity, duration, or parameters of ADSR, LFO and filters. We therefore classify this mapping as a “many-to-many”.

Aiming for greater similarity to the controls used in the game, the eight main buttons were used in a similar way to the first mapping. The directional buttons, when just pressed and held, change the sound in a simple way.

The “hold forward” action in games means moving toward your opponent (assuming you’re on the left side of the screen, which is common for player 1 at the beginning of the fight). This results in stronger sounds with short

ADSR envelope and without the LFO to symbolize simple blows.

Holding the back button would mean defending or retreating your character. We chose to vary the message channel in order to activate another synthesizer with percussive sounds for that case. If any of these buttons is pressed twice in a row, it activates the LFO in a way that the sounds will be executed in a quick succession. In the game those commands would make the characters to dash.

Up, which causes the character to jump, means to go up an octave while down, which would be to crouch, means to go down an octave. These commands cam be used in the same way described above with the same effect. The diagonals can be use to achieve the results of pressing the combination of directions, like up and forward at the same time. For the more complex commands, an analysis was performed under the list of possible combos of the game Street Fighter IV. In this way, the most common commands used in the game were grouped in the form in which each is responsible for generating a specific effect.

Instead of thinking on isolated parameters we started thinking on abstract concepts, like force and energy, to determine the effects generated in each sound event. That is, the longer the command, the more intense and long-lasting the result will be.

If we look at the list of commands it is easy to see that although they vary from character to character, there is a pattern between them. Among the most common are the half-circle and quarter-circle that can be performed backward or forward. Such commands require the player to move the joystick or slide the finger across the D-pad to complete these patterns. Some commands require the player to hold the joystick in one direction and move in the opposite direction quickly. There are more powerful attacks called Ultra Combos that often require the above moves to be made in sequence.

For this mapping we added the execution of glissando. Holding the joystick back and then moving it forward will result in a crescent glissando. This movement can be performed in succession to increase duration. If the movement is performed from bottom to up it will have the same effect. Such movements performed in the opposite direction will result in an decreasing glissando.

The quarter circle results in the execution of chords that vary according to the button pressed at the end and whether it was executed clockwise or counterclockwise. If executed twice then it increases the dynamics and duration of the envelope.

A half-circle will perform these chords in increasing glissando. Just as before, running them in sequence will increase their duration. To perform a decreasing glissando, it is necessary to finish the combo on the D-pad with the direction opposite to the one half circle was executed.

All commands always end by pressing one of the main buttons. Which of these has been pressed will determine the pitch of the resulting sound. To vary the octave in this way is done pressing the D-pad or moving the joystick down or up before the combos. An alternative is to use the buttons on the left and right joysticks to vary the octave in a fixed way. The organization of the last mapping is illustrated in the table 6.

Table 6: Many-to-many mapping using video game combos

Buttons	Mapped to
Button 1 to 8	NOTE activation
Left Analog	Octave down
Right Analog	Octave up
Start	Panic Button
Axis	**Mapped to**
D-pad	Combo entry
Left Analog	equals D-pad
Right Analog	Volume

4 Discussions

Despite all the possibilities described in our mapping implementations, it is possible to argue that DualAnalog and our project and implementation does not bring great advances in relation to the standard computer entries, like keyboard and mouse, in relation to musical expressiveness. The set of discrete buttons provided by the control are as limited as a mouse button since they do not give information about the applied force and give just the number of the clicked button. On the other hand the way the controller and the arrangement of the buttons on it can be handled top allow a faster and intuitive access to them for those used to this ind of interfaces.

In our opinion, these interfaces have some advantage if compared to a keyboard and a mouse that is the set of continuous axes. The pair of joysticks allows the simultaneous control of 4 continuous axes, which can only be done with two axes through a mouse. However, this control can only be done from a self-centered input value, giving always a relative value and not an absolute value. For this reason, it is impossible to leave the joysticks stopped at a specific position and a continuous movement is necessary to deal with this interface. In fact, this is not a complex interface that requires training to be used satisfactorily.

Returning to the original context of this interface, we will analyze the interaction in an electronic game between the player and the system. To beat the game, one will need not only an understanding of the challenges posed, but also how to use the controls made available efficiently. So, even if the joystick itself is a simple interface, mastering its use in a particular game may require some training. This is related to the way in which we choose to map commands and can influence interface performance.

The video game control we use also has some drawbacks. Analogs are too sensitive to perform a more precise movement. For the choice of mappings some physical impediments must be taken into account: it is not possible to manipulate the D-pad at the same time that the left analogue is used; it is not possible to reach the 4 front buttons at the same time the right analogue is used and the simultaneous use of many buttons at the same time requires training.

5 Final Considerations

In this paper we presented the possibility of using video game control for musical interactions focused on the different mappings to a synthesizer. More specifically, we worked with a variation of the DualAnalog control developed by Sony that has a USB input to be connected to personal computers. The connection between electronic games and music comes from the beginning of the video game era [14]. We believe that this is a point that can be explored in the creation of musical performances.

The task of developing mappings, taken as the focus of our research, brings up a unlimited range of possibilities. This also has its negative side, since it is easy to get lost in such a vast horizon. Finding mappings with strong semantic meanings is not a trivial task, especially in the case where there is no acoustic instrument to be taken as a reference [6]. So, for our more complex mapping we made the choice to base ourselves on the universe of electronic games.

Our research aims not only to create a digital musical instrument, but the elaboration of alternative controls to be used in the general artistic context. We chose to use MIDI protocol and compatible synthesizers because of their great popularity and due to the ease of prototyping. However, we believe that the implementation of a proprietary synthesizer brings greater control over sound and increases the number of synthesis parameters to be considered. Still, we have had good results with the choices we made. We were focused on the mapping and we believe that all these proposed mappings can be be used with other synthesizers.

The various mapping modalities presented, with their respective examples, serve as a basis for future research. As we have argued, there are numerous possibilities for mapping and each can result in a digital musical instrument with very different characteristics.

The choice of using MIDI synthesizers is justified, but can be discussed. MIDI is a Curse, is still here, and we still use it. But Open Sound Control (OSC) became a great option too[15].

6 Acknowledgments

Authors would like to thanks to all ALICE - Arts Lab in Interfaces, Computers, and Else members that made this research and development possible. The authors would like also to thank the support of the funding agencies PROPE, CNPq and FAPEMIG.

References

[1] Marcelo M Wanderley and Philippe Depalle. Gestural control of sound synthesis. *Proceedings of the IEEE*, 92(4):632–644, 2004.

[2] Vesa Välimäki and Tapio Takala. Virtual musical instruments—natural sound using physical models. *Organised Sound*, 1(2):75–86, 1996.

[3] Gianpaolo Borin, Giovanni De Poli, and Augusto Sarti. *Musical signal synthesis*. Swets & Zeitlinger BV, 1997.

[4] Joel Ryan. Effort and expression. In *Proceedings of the International Computer Music Conference*, pages 414–414. INTERNATIONAL COMPUTER MUSIC ACCOCIATION, 1992.

[5] Andy Hunt, Marcelo M Wanderley, and Ross Kirk. Towards a model for instrumental mapping in expert musical interaction. In *ICMC*. Citeseer, 2000.

[6] Andy Hunt, Marcelo M Wanderley, and Matthew Paradis. The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, 32(4):429–440, 2003.

[7] Andy Hunt and Ross Kirk. Mapping strategies for musical performance. *Trends in gestural control of music*, 21(2000):231–258, 2000.

[8] Matthew Blessing and Edgar Berdahl. The joystyx: A quartet of embedded acoustic instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 271–274, Copenhagen, Denmark, 2017. Aalborg University Copenhagen.

[9] Sook Young Won, Humane Chan, and Jeremy Liu. Light pipes: A light controlled midi instrument. In *Proceedings of the 2004 conference on New interfaces for musical expression*, pages 209–210. National University of Singapore, 2004.

[10] Christopher Ariza. The dual-analog gamepad as a practical platform for live electronics instrument and interface design. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan.

[11] Jace Miller and Tracy Hammond. Wiiolin : a virtual instrument using the wii remote. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 497–500, Sydney, Australia, 2010.

[12] Elaine L. Wong, Wilson Y. F. Yuen, and Clifford S. T. Choy. Designing wii controller: A powerful musical instrument in an interactive music performance system. In *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '08, pages 82–87, New York, NY, USA, 2008. ACM.

[13] Joseph Butch Rovan, Marcelo M Wanderley, Shlomo Dubnov, and Philippe Depalle. Instrumental gestural mapping strategies as expressivity determinants in computer music performance. In *Kansei, The Technology of Emotion. Proceedings of the AIMI International Workshop*, pages 68–73. Genoa: Associazione di Informatica Musicale Italiana, October, 1997.

[14] Ben Olson. Transforming 8-bit video games into musical interfaces via reverse engineering and augmentation. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, volume 16 of *2220-4806*, pages 73–77, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[15] Perry R. Cook. Re-designing principles for computer music controllers : a case study of squeezevox maggie. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 218–221, Pittsburgh, PA, United States, 2009.

TumTá and Pisada: Two Foot-controlled Digital Dance and Music Instruments Inspired by Popular Brazilian Traditions

João Tragtenberg[1 2] **, Filipe Calegario**[1 2] **, Giordano Cabral**[2] **, Geber Ramalho**[2]

[1] CIIMUS / Instituto SENAI de Inovação para TICs / SENAI-FIEPE
Rua Frei Cassimiro, 88 - Santo Amaro - 50.100-260 - Recife - PE

[2]MusTIC / Centro de Informática / Universidade Federal de Pernambuco
Av. Jornalista Aníbal Fernandes, s/n - Cidade Universitária (Campus Recife) - 50.740-560 - Recife - PE

`tragtenberg@gmail.com, fcac@cin.ufpe.br`
`grec@cin.ufpe.br, glr@cin.ufpe.br`

Abstract. *This paper presents the development process of "TumTá", a wearable Digital Dance and Music Instrument that triggers sound samples from foot stomps and "Pisada," a dance-enabled MIDI pedalboard. It was developed between 2012 and 2017 for the use of Helder Vasconcelos, a dancer and musician formed by the traditions of Cavalo Marinho and Maracatu Rural from Pernambuco. The design of this instrument was inspired by traditional instruments like the Zabumba and by the gestural vocabulary from Cavalo Marinho, to make music and dance at the same time. The development process of this instrument is described in the three prototyping phases conducted by three approaches: building blocks, artisanal, and digital fabrication. The process of designing digital technology inspired by Brazilian traditions is analyzed, lessons learned, and future works are presented.*

1 Introduction

"TumTá"[1] is a wearable instrument in the form of a pair of insoles to be placed inside the shoes. The gestural design of the instrument was focused on the foot-stomping gesture with the heels. The mapping is straightforward, relating the stomping to the triggering of sound samples; the sound volume depends directly on the intensity of the stomp. The selection of the samples to be triggered can be done through an auxiliary interface called "Pisada", which consists of a series of pads distributed across the floor to be stepped on.

The development process has taught some lessons about the development of Digital Dance and Music Instruments (DDMI) [1, 2], by combining the expression through body and sound at the same time. Another exciting aspect of this process was the artistic context that guided its development. Instead of following aesthetic principles from European hegemonic cultures, a common bias in most of the music technology research and development, this process was based on grassroots popular Brazilian traditions [3]. This aspect has a lot to contribute to the Computer Music area and may be best suited for musicians and dancers based on similar traditions.

[1]Videos available on `https://youtu.be/m4q6iD513pY` and `https://youtu.be/d0F54uO15-w`

Figure 1: TumTá's insoles and transmitter hardware

2 Motivation

The "TumTá" started from a demand of the artist Helder Vasconcelos of an instrument that produced sound from dance gestures with a direct relation. The development of an instrument for music and dance was quite challenging. We found that the bibliography mostly differed between Digital Musical Instruments (DMIs) and Interactive Dance Systems (IDSs) and few references shared a perspective for expression through both art forms at the same time. The concept of Digital Instruments of Dance and Music (DDMIs) was created during this research process [1].

The development of "TumTá" began in 2012 and featured several stages of prototype development with frequent evaluation meetings with Helder Vasconcelos. The design process was focused on the use of Helder and guided by his artistic needs, without having been tested by any other user until 2017. This whole process happened without any funding, concomitantly with other works, and without a research laboratory infrastructure.

Figure 2: Helder Dancing/Playing "TumTá" during his performance

2.1 Helder Vasconcelos

Helder Vasconcelos is an artist renowned for his work as a musician, actor, and dancer. His formation took place in the *brincadeiras* of Cavalo Marinho (a famous theater from Pernambuco) and Maracatu Rural (tradition linked to the carnival of Pernambuco). He acted as a musician playing percussion instruments and eight-bass accordion in the band Mestre Ambrósio, where he also danced and performed characters during concerts. He also created two solo performances: "Espiral Brinquedo Meu," where the primary language was theater, and "Por Si Só," mainly driven by dance. In this second solo, he used Interactive Dance Systems created by Armando Menicacci [2] to produce sounds and projections from his dance.

From the experience of working on another solo with digital technology, Helder already had a clear guideline for the technologies he was looking for: autonomy of use. During his second solo, the sensors and cameras he used to dance to interact in real time with several people were replaced by pre-recorded media because he couldn't use them by himself. He had also a specific demand for the functional requirements of his instrument and how he would make use of that of the device:

"I wanted very objectively a MIDI trigger on the heel that when I hit it on the floor it gave me an information, because all the rest I already had in my head [...] in a nutshell I wanted a deeper sound on the right foot and a sharper sound on the left foot, that was exactly the relation of the *zabumba*: the *bacalhau*[3] in the left hand and, if you are right handed, and the mallet in the right hand. I wanted to reproduce exactly that on my right foot and on my left foot, which is precisely the *zabumba*'s 'Tum' and the *bacalhau*'s 'Tá'. So that's why the instrument ended up with that name, I wanted a 'Tum' on the right foot and a 'Ta' on the left foot. Although I already glimpsed several things "(Interview with Helder Vasconcelos)

This reference to the *Zabumba*, an instrument widespread in many genres of the North-eastern Brazilian music, shows how "TumTá" clearly presents an instrumental inheritance[4]. The idea of an instrument that can trigger only two percussive sounds, one high-pitched and another low-pitched could be seen through hegemonic traditions as an instrument without much expressivity or diversity [5], but in the context of popular Brazilian music and many other traditions there are many instruments pervasive to many genres that are undoubtedly expressive and diverse with just two classes of sounds.

This search for expression both musical and corporal is a recurrent element in the trajectory of the artist and he credits his formation to those traditions, where this relationship is latent. Helder Vasconcelos systematized his learning process in these traditions in principles for making music, dance or theater. His principles are based on the idea that a gesture or a sound starts from a common essence, which he calls the generating impulses:

"The generating impulse of something that can become a dance, a song or a theater is the same. So I do not see it separately. It will all depend on the artistic necessity. What do I mean? What do I want to say? What do I want to build? [...] This creative impulse can become a character, it can become a choreography, or it can become a beat, a rhythm, or a song. [...] I don't see them as three integrated things or three separate things. In the place of the creative impulse I don't think what it will be." (Interview with Helder Vasconcelos)

From this theoretical background, Helder sought an instrument capable of revealing this common essence of gestures and sounds, and he was seeking for this instrument to instantly translate the impulsive quality of the impact of the footfall on the ground with the impulsiveness of the sound produced. Our challenge was to develop a device for perceiving in real-time the impulsiveness of a foot stomp and mapping it into sound.

2.2 Artistic References

As well as searching for an instrument with an instrumental inheritance of the Zabumba, the choice of this gesture of foot stomps with the heel had an inspiration in the dances of *Cavalo Marinho*, that many dance steps share this bold stomping:

Figure 3: Totó Teles and Cavalo Marinho Estrela Brilhante from Condado(PE) - Brazil *brincadores* dancing a *trupé*

"This expression in Maracatu and even more in Cavalo Marinho, it is in the contact of the foot with the ground. We call it 'Trupé', the dance, the movement. It is a very percussive movement very naturally between the

[2] Armando Menicacci is a research choreographer and professor of new media for interdisciplinary performance at Université di Quebec à Montréal (UQAM) and one of the first users of the "Isadora" programming environment.

[3] The *Zabumba* is a Brazilian percussion instrument that is played with a mallet and thin stick called *bacalhau*

foot and the floor. So it comes from my school from the traditions that I participate in. This is already like this, so it was just a matter of empowering, saying, 'what if it actually made some sound out of that?'" (Interview with Helder Vasconcelos)

The steps of Cavalo Marinho have the characteristic of marking a rhythm very characteristic and common to several rhythms of Brazilian popular music. It is the same rhythmic cell of the *baião*, that usually is played by a *zabumba*:

"In several *trupés* of Cavalo Marinho we already do that [playing baião rhythms], even without making that sound, but we already play the zabumba thing. The Baião, the classic form of 'Tum Tum Tá Tum Tum' you play it with your foot, even without sound, dancing Cavalo Marinho. That's why it was so objective for me: 'If I have this, it will appear' "(Interview with Helder Vasconcelos)

This tight connection with a cultural tradition contributes to the degree of instrumentality [6] of "TumTá", allowing it to be perceived by musicians as a musical instrument and to create a pact with the audience [7] on understanding what sounds are connected to which gestures. These aspects are a rare situation for a new instrument, with which usually the audience does not know what to expect from the interaction. A dance with rhythmic characteristics of the foot contact with the ground strengthens the recognition of the dance made with the instrument and incorporates a gestural repertoire of this tradition. Several other tap dance traditions already carry an intimate relationship of percussive musical creation with footsteps, which brings to this mapping strategy an attractive potential for expressiveness.

3 Similar Instruments

3.1 Miburi

The Yamaha Miburi was an interactive whole-body wearable system with several sensors distributed throughout the body. Japan's Yamaha Experimental Division launched it in 1994. The system consists of a blouse with six bending sensors in the arm joints, two hand controllers, each with ten pressure-sensitive buttons and continuous controllers for the thumbs. The system also included two adjustable sized insoles with piezoelectric heel and toe sensors that detect tapping gestures. All these sensors were connected with a cable to a unit used as a belt. This unit communicated wirelessly (only for the version sold in Japan, for the exported versions only a cable connection was possible) with a sound synthesizer unit that had its amplifier.

3.2 Expressive Footwear

"Expressive Footwear" was developed by the Responsive Environments Group, coordinated by Joe Paradiso of the MIT Media Lab. Its interface consisted of a pair of snickers with 12 sensors that continuously send 16 raw data on the movement of the feet. It was developed to capture all the gestural possibilities of the feet like flexion, inclination, position in space in three dimensions, turns, orientation, pressure against the floor and foot stomps with the heel or tip of the foot.

The instrument was inspired directly by the Miburi's foot module during a visit of Joe Paradiso to the Yamaha experimental division. He claimed to have been inspired by its insole, seeking to overcome its limitation by adding sensors for continuous feet gesture recognition. The main guidelines were to eliminate the wires that connected the Miburi's insoles to the central unit, keeping all electronics wearable on the footwear. He sought a perception of continuous qualities of foot movement rather than discrete shots, which increased the possibility of interaction for free foot gestures and gestures of interaction with the ground [8].

3.3 Other Similar Instruments

Several other instruments have been developed for foot control beyond Miburi and Expressive Footwear. An example cited by Paradiso is a system with piezoelectric sensors for tap dancers [9]. Jônatas Manzolli developed a similar device in 1997 for the performance AtoContAto with the dancer Christiane Matallo [10].

4 Development Process

The development of "TumTá" started with the simplest and fastest solutions to trigger sounds from foot gestures. The prototypes were gradually evolving and acquiring increasing complexity and robustness. In order to overcome challenges, physical and functional prototypes were created to enable a practical evaluation of each solution proposition. The collaborative evaluation by the team brought up positive and negative points of each solution, and the discussion directed the approach we should take for the next prototype.

"Pisada" was developed to be an accessory for changing "TumTá" sample pairs, but it followed a parallel development process and ended up serving also as an independent Digital Dance and Music Instrument, serving for much more purposes. That is why we will present it in a separate section of this paper.

We started off the project with quick and dirty prototypes using off-the-shelf technology from game controllers (Nintendo Wiimote and Microsoft Kinect). Deciding on the approach of wearable sensors, we used an Arduino with some shields for connecting a DIY pressure sensor and an XBee transceiver to a Pure Data patch through an artisanal approach for the electronics and case. After validating the idea, in the next prototyping phase, we improved the robustness and size of the system using digital fabrication tools, designing a custom PCB and laser-cut enclosure, a better transceiver and a more user-friendly software interface.

These three approaches using ready-made building blocks, hand-crafting, and with digital fabrication [11] have shown to be an exciting design process. We used each process' advantage in favor of each part of the process. These approaches were analyzed in the development of the Prosthetic Instruments, which showed some similar challenges we faced creating a new instrument for the specific performance of artists that were not the hardware developers.

In the subsections below, we described in further details the challenges and advantages we face on each part of the process.

4.1 First Prototyping Phase: Building Blocks

In the same meeting, some researchers from MusTIC research group, we used off-the-shelf building blocks prototypes that were assembled on the same day. Jerônimo Barbosa presented a prototype made with Microsoft Kinect where the quantity of movement of the whole body activated a state machine. This prototype did not have a direct relationship between gesture and sound, but a more abstract connection and was not connected to what Helder was seeking. The other prototype presented was with a Wiimote of the Nintendo Wii console by the second author. The control was taped to the dancer's ankle kand a simple mapping strategy with OSCulator [4] was created to trigger MIDI notes when the sensor's accelerometer data passed a threshold. Despite the high latency and size of the controller, this prototype brought a positive reaction in Helder:

"Since the first test with the Wii tied with masking tape, it was very, very, very stimulating. I already felt something happening [...] that first test already had a stimulus to say 'if it works I'll use'" (Interview with Helder Vasconcelos)

This Wiimote solution represents a widespread approach, having been one of the standard interfaces of the NIME community [12]. This first stage made clear the artist's choice for a wearable system, which was an essential constraint for development. We discarded the option of the camera-based solution because of many factors, for example, the need for lighting calibration, the possibility of occlusion, or the high latency of the cameras.

Even though the building blocks were not as small and with a fast reaction, it was enough to validate the idea, which leads us to improve the next prototypes in size, sensing accuracy and gesture-to-sound latency.

4.2 Second Prototyping Phase: Artisanal

Once established that the instrument would be a wearable device, we began to develop more responsive prototypes with a clear goal of having a lower latency. The first prototype craft was made with a pressure sensor (FSR 402 from Interlink) connected to a protoboard and an Arduino UNO. Every time the pressure went beyond a threshold, it triggered a MIDI note that in a DAW triggered a sound. We performed tests by tapping the sensor with a finger on a table and perceived outstanding responsiveness, without perceiving any latency.

Nevertheless, when we did the first tests with the sensors inside the shoes, the sensors broke on the first week

[4]https://osculator.net/

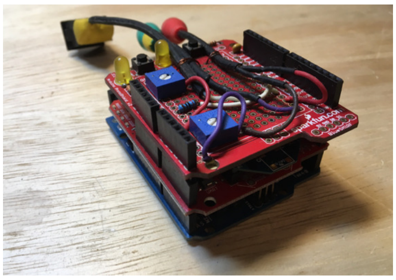


Figure 4: "TumTá's" second prototype made out of arduino uno with shields (to be enclosed in a travel soap case)

of tests due to the fragile structure of the sensors and heavy pressure load of the foot stomps. We also tried a piezo sensor, but no matter how much hot glue we put into protecting it, it always broke on the first stomps. The ruggedness of the wearable system, especially for dancing, is a great challenge for the hardware developer. Similar problems were found by the MIT Media Lab team while developing the Expressive Footwear: "A dancer's foot is indeed a hostile environment for sensitive electronics, and as we have been reminded repeatedly by experience, everything needs to be well attached or latched down - anything that can move will sooner-or-later break off" [8]

We migrated to a do-it-yourself sensor alternative with conductive foam that the XBee radio had come. This alternative was inspired by a tutorial published on the Instructables portal by Kyle McDonald [13]. This prototype showed a much less accurate reading than the purchased sensor and presented a very noisy signal, but it was very robust at impact.

Figure 5: DIY sensor made out of a conductive foam insole and conductive thread electrodes

The next prototype was made from out of an antistatic insole made of conductive foam. We used conductive stainless steel (Bekinox VN 12 / 2X275 / 175S / 316L from Bekaert) for the electrodes, which presented an excellent resistance to impact. Sensors previously created with small pieces of conductive foam had the problem of sliding during use, a problem solved when the sensor was shaped as an insole by fitting perfectly inside the shoe. The whole stack Arduino Shields[5] was enclosed inside a plastic trav-

[5]This artisanal approach also used Arduino Shields building blocks, instead of using custom soldered breadboards, this accelerated a lot the hardware development process

eler's soap-dish, fixed with an elastic bandage around the waist.

Another challenge for the development of wearable instruments that we found refers to the electrical noise of the human skin. We noticed a noise in the sensor signals when the foot came in contact with the sensor. It was necessary to coat the insole with a rubber (neoprene) to isolate the circuit of this noise. Any solution with a conductive line, lacking electrical insulation, is subject to these noises.

Even with several problems in these first prototypes, it gave a much better solution than the taped Wiimote, allowing him to give his first presentation to a broad audience at "Mostra Rumos" in São Paulo's Itaú Cultural main theater, part of the Rumos Itaú Cultural dance award for 2012-2014. The artisanal solution did not allow an autonomous use, needing of a specialized professional to fix software and hardware bugs which were still frequent. This approach was very customizable but gave unstable solutions that could be improved in ruggedness, size, and user-friendliness.

4.3 Third Prototyping Phase: Digital Fabrication

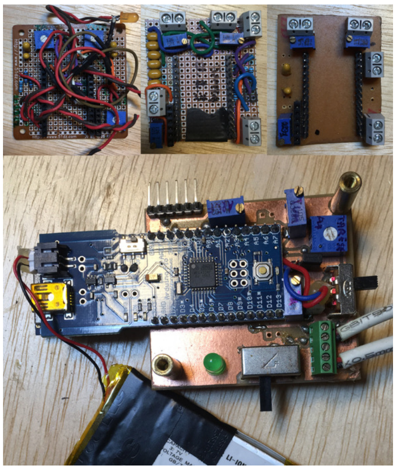


Figure 6: Sucessive versions of the prototype's PCB, improving it's quality and reducing the amount of wires

With the arrival of Fablab in the city of Recife, we had access to a laser cutter, with which we were able to take advantage of techniques of digital fabrication. This approach enabled us to create prettier and better enclosures with greater precision and robustness. We also started manufacturing our PCBs with the laser-cutter, which lead to much more rugged circuits that did not present bad contact issues from fatigued wires. The new circuits went through several stages of development (Figure 6), seeking a smaller size and a more trustworthy circuit with fewer wires.

We also had several problems with the wireless transmission. The XBee s1 module had a good range of places without many 2.4GHz devices (Mainly Wifi and Bluetooth devices), but as the noise grew, the shorter was the range it worked. For a fast solution, we changed to an equivalent more powerful transmitter (XBee Pro S1) to have a strong signal even in electromagnetically noisy places. Packet loss was somewhat inadmissible for the low fault tolerance that the condition of artistic presentation demands.

The stomping recognition used an algorithm that the first step was responsible to store the sensor data in a sliding window with the last five readings. A median filter averaged these five values to reduce noise from the sensors. From the last five averages, a derivative approximation of the data was calculated with a backward finite difference algorithm. The values of the derivatives were monitored through a Noise Gate filter, only to be considered when they passed a certain threshold. When a maximum local value was found, an event was triggered, and that value was related to the intensity of the stomp. As the data presented at least two peaks for each stomp, we implemented a debounce algorithm to only consider the first one in a time interval of 150 ms. This interval was much bigger than the regular interval between bounced peaks, but since it was not possible to give two consecutive stomps in a time shorter than that we kept the highest stability of the instrument.

The use of the pressure's rate of variation (numerical derivative) to detect the stomps allowed them to be differentiated from simple steps. When walking, the pressure ranges from minimum to maximum values as the weight is transferred from one foot to the other, but the pressure variation in each foot occurs very slowly. The detection threshold allowed to incorporate the range in which the derivative of the pressure corresponded to basic steps, allowing the user to walk without triggering any sound. The stomps occur with a higher pressure rate variation, and the adjustment of this threshold was made with tests of walking steps against stomps.

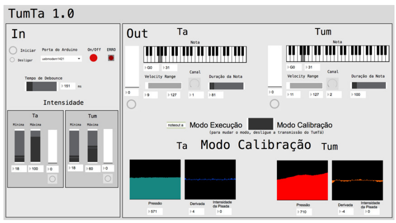


Figure 7: Max/MSP patch for mapping the sensor data into MIDI note triggers

For improving the usability, a Max/MSP patch was developed with an improved graphical user interface, allowing Helder to quickly understand how to calibrate his instrument and configure simple parameters like the dynamic range and MIDI notes and channels each foot would trigger.

After almost three years of development, we came to a stable version of "TumTá". With this version, Helder Vasconcelos used it in the debut of his third solo "Eu Sou." It was possible to have a dynamics control between notes of weak, medium, or strong intensities with few nuances between them. The solo debut took place at the SESC Pompéia theater in São Paulo as part of an occupation for a month, where the artist also presented his other two solos. The dance and music made throughout the show with "TumTá" allowed much freedom for improvisation and control over the execution.

4.4 Future challenges

The main technical problem during development was related to the UART serial protocol used to read data in Max/MSP from the XBee Receiver. After the last version, we noticed that the standard UART baud rates (9600, 19200, 38400, 57600, 74880, 115200 bps) have a considerable amount of dropped packets. This issue is not related to regular devices, but to musical applications they are inadmissible. The low priority of the FTDI[6] driver reading USB serial data in the operating system of the computer may have also contributed in no small latency, jitter, and unrecognized stomps.

Recent studies have shown how solutions with Arduino communicating over serial ports with Max/MSP are unable to meet the high latency and jitter requirements of percussive musical instruments [14]. The authors suggest that the MIDI protocol should be used whenever possible. Even though the baud rate of 31250 bps is relatively low, it synchronizes with the microcontroller's clock value and does not drop any packet. It adds to that reason that the MIDI device USB drivers are optimized for a higher priority in the OS, considering its time sensitiveness. The common critiques around the limitation of MIDI messages to be limited to 7 bits (from 0 to 127) are not precise since the MIDI protocol can use Non-registered parameter Numbers (NRPN), system Exclusive messages (SysEx) or other methods to send continuous data with a more extensive bit range.

Some even more recent research shows that Open Sound Control (OSC) and MIDI Over Bluetooth Low Energy (BLE) protocols with the readily available ESP32 microcontroller can deliver data with a latency below the 10ms accepted standard without the need of a hardware receiver [15].

Due to the artisanal nature of this sensor, it is hard to replicate it into similar products. The lack of sensitivity of the instrument for weaker foot stomps was a challenge that remained unsolved in this version because of the noisy signal from the sensor and represented the most noticeable problem from the users that tested it.

"Its sensitivity is so important [...] today, it is hard, he demands a lot. [...] it requires a certain posture and a certain pressure, a certain force for it to trigger the sound. [...] so it still has this tension [...] not triggering is very frustrating. It disorganizes a lot. It is as if you are going to play a note, you are going to play chord a piano and it didn't trigger any sound. [...] making it more sensitive, more precise in this way, that would also let me loosen up, it would increase the possibilities, because I would not be in this place of tension: 'Jeez, maybe it won't trigger anything" (Interview with Helder Vasconcelos)

[6] chip used to convert RS-232 serial protocol to USB

Another factor to be rethought is the need for a computer to operate the instrument; that is, we want the next versions to have a hardware sound unit to trigger the sounds. This factor would facilitate its use and logistics, allowing presentations to informal contexts and alternative places like squares, streets, etc. Currently, in addition to relying on the computer, it requires specific software that may be deprecated in future versions of the OS. It needs to have hardware that can have a 1/4" jack for triggering its samples and a USB and DIN MIDI output connector for more excellent compatibility with low latency with other gear.

5 Pisada

Figure 8: "Pisada" pads spread on the floor

"Pisada" is a DDMI designed initially for changing the sound banks of "TumTá" while dancing. Its sensor interface consists of ten square pads of 25x25 cm connected to a hub through, to be pressed with the feet. It holds the same functional principle as a regular MIDI pedalboard. The main difference is the size and structure of the buttons, that are larger and spreadable around the stage, imposing fewer restrictions to body movement while pressing it.

Since it is a MIDI controller with ten buttons and ten pages that change the function of each pad, it can serve for much more than a "TumTá" accessory. It allows several musical controls in DAWs and Synthesizers such as turning on and off effects, triggering sounds or selecting tones.

A traditional MIDI pedalboard usually has several buttons separated by a few centimeters from each other, requiring the user to stand in a region of space and make a slow and careful movement to push each button. The "Pisada" allows the control gesture to be anywhere on the stage and pressed during a dance without a cognitive restriction on the performer's movement.

The "Pisada" works as an incredibly safe, accurate and straightforward to use position sensor. It can detect a region with a radius of up to 10m, by the length of its wires (which can be extended). Unlike computer vision techniques that require complicated procedures for adjusting brightness and configuring the division of space, "Pisada" can detect when the artist is in some position with effortless installation and use procedure. By having tactile feedback and being a visible object, it allows much more precise temporal control than camera position detection systems.

Beyond the initial function to alter the samples triggered by "TumTá", Helder Vasconcelos used it in his show "Eu Sou" in various ways like triggering playbacks for songs, activating and deactivated effects on his microphone, for recording live loops, triggering live recorded loops and activating or altering functions of other digital instruments like a tap-dancing floorboard with Tet Music's "Pulse Controller" [7]. "Pisada" was the instrument that placed the controls which are usually on a backstage technician's hands, on the artist's feet.

It was made in an artisanal way with two galvanized iron sheets with rubber tabs on the corners. When it was pressed, the plates touch and close a contact measured by the microcontroller. When stepping it sends a note on signal and release sends a note off signal. There is only a simple debounce algorithm with a relatively high value of 300 milliseconds to ensure that a button is pressed once. They are all covered with black linoleum to be discreet on traditional stages.

In addition to the ten squared pads, two triangular pads are responsible for changing the button-to-MIDI-events mapping, in a metaphor of pages. The ten pages are shown in a seven segments LED display, allowing 100 different notes to be triggered. There is also visual feedback LED on each rectangular pad that lights up when pressed. The ten pads are connected to a central hub by long RJ45 cables of 10 meters, and this hub can be connected to a computer by a MIDI USB cable.

5.1 Future challenges

The instrument's simplicity made it very easy to use, but the artisanal process of building it takes too much time and is too much expensive since it uses thick iron sheets. It was also too heavy and presented serious airplane shipping problems. The cables represent a problem because they need to be dispersed on stage, and the performer cannot step on it, only allowing them to be positioned on the edges of the used stage area.

The instrument would benefit a lot by a changing in the material and structure to a lighter and more straightforward to make a structure and a robust wireless connection would surely be a much better alternative. Long cables are costly, with reasonably short durability and require much time to be positioned in space. It may also be interesting to add sensitivity to the pressure to afford new control possibilities.

The "Pisada" is a simple instrument developed with the premise of bodily expressiveness for precise musical control. It was designed to be robust, easy to learn, to have a low latency and precise responsiveness. It presents a straightforward, low cost and robust solution for the detection of position in space.

[7] http://tetmusic.com/

6 Sound Design

The sound design of "TumTá" was all made in Ableton Live using its "Simpler" sampler instrument. The sampling technique was used for the simplicity and precision to trigger sounds from acoustic instruments from popular traditions and other electronic sound samples. Helder Vasconcelos triggered in his home studio his Bombos (an instrument similar to a zabumba used in his Boi Marinho tradition), Preacas (instrument from the tradition of Caboclinho, which consists of a percussive bow and arrow), a Kaïamb (a shaker from Réunion Island). Some synthesizer samples were also played and also samples of his voice.

The samples were recorded both as one-shot sounds and parts of a loop playing a baião rhythm with the Bombo. In some sample banks, both feet were mapped to the same one-shot samples. On other banks, one foot was mapped to the *bacalhau* stick of the bombo, giving a high pitched sound, while the other foot was mapped to a part of a loop that needed to be retriggered in precise time to keep the rhythm going in the beat. Specific dance steps made rhythm variations retriggering the sample in shorter periods. The kayamb was triggered in a looped pattern with a sizeable decaying amplitude.

It was mapped in a way that each new "Pisada" pad pressed changed the samples. The group of samples that played loops were built in a way that new banks repeated the last samples with new layers added to it, in a similar way to the structure of Electronic Dance Music, allowing him to give more liveness to a performance that a building up rhythm with superposed layers of sound timbres.

6.1 Future challenges

The versatility of a sampler was the right choice for the straightforward sound design based on traditions that rely only on acoustic instruments. A physical modeling synthesis would also be an exciting approach, giving more subtlety in random sound variations, typical from percussive acoustic instruments, but would lead to much more time and specific knowledge to be developed. This approach of allowing sound variations depending on velocity and on random parameters could also be used in the "TumTá" sampling engine with many samples per foot, and that would undoubtedly make the instrument a lot more expressive. The sampler also gives the instrument an exciting potential for the instrument to be adopted by other musicians and dancers, for the simplicity of customization and sharing of sound banks.

In literature, Sergi Jordà presents the concept of Diversity, that helps to define the concept of Music Instrument Efficiency [5]. The instrument's diversity is subcategorized in Micro-diversity, related to performance nuances (how two performances of the same piece can differ), Mid-diversity, related to performances' contrasts (how distinct two pieces played with the same instrument can be) and Macro-diversity related to stylistic flexibility (how the instrument adapts to different contexts). From this perspective, “TumTá” could easily be considered a not very diverse instrument, from a simplistic perspective that it needs to have many controls to be diverse.

“TumTá” was designed to be as diverse as a Zabumba, that even though it can trigger small variations of two sounds, it has a high diversity in all dimensions. The current lack of diversity of “TumTá” is related to its low dynamic range of control, only detecting foot stomps that are too hard. An improvement in the sensor interface and better sound design with a more expressive sampler would improve a lot its diversity and therefore, its efficiency.

7 Conclusions

This paper presents “TumTá” and “Pisada.” Two original Digital Dance and Music Instruments inspired by Brazilian popular traditions. This process pointed out was how music and dance traditions could influence the development of new technology and is a rich source of inspiration for the design of original NIMEs. The development process of the instrument was described in technical details, following three main prototyping phases that overlapped each other, but each had a specific approach on Building Blocks, Artisanal, and on Digital Fabrication. Both future instruments challenges were presented separately and had a lot to improve, but are already used in professional artistic contexts with success.

8 Lessons Learned

From the development process, many technical lessons were learned from specific challenges in each prototyping phase:

- Building Blocks are a good approach for a first prototyping phase. Even though they are not customizable, it gives quick answers to major decisions.
- The Artisanal Approach is interesting for a highly customizable solution, giving freedom to have original ideas while is time-consuming, not easily reproducible and can lack reliability, precision, or ruggedness.
- Digital Fabrication techniques can shorten a lot the design process giving precise and reliable solutions with highly customizable prototypes. Nevertheless, the possibilities are a lot limited by the machine’s constraints.
- Wearable devices for dancers need to be EXTRA RUGGED, and needs isolation because of the sweat.
- Wireless transmissions need to be powerful for long distances (beyond the noise around you)
- MIDI and OSC are always better solutions than custom serial protocols; there is no need for reinventing the wheel.
- Sample-based synthesis is great for percussive instruments and an accessible way to create sounds with a Brazilian musical identity.

References

[1] João Tragtenberg, Filipe Calegario, Giordano Cabral, and Geber Ramalho. Towards the Concept of Digital Dance and Music Instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Porto Alegre, Brazil, 2019.

[2] João Tragtenberg. Instrumentos Digitais de Dança e Música: Uma Proposta de Paradigma para Potencializar o Design de Instrumentos para Expressão Musical e Corporal MSc. Thesis UFPE, 2017.

[3] Jerônimo Barbosa, Filipe Calegario, João Tragtenberg, Giordano Cabral, Geber Ramalho, and Marcelo M Wanderley. Designing DMIs for Popular Music in the Brazilian Northeast : Lessons Learned. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 277–280, 2015.

[4] Filipe Calegario. *Designing Digital Musical Instruments Using Probatio: A Physical Prototyping Toolkit*. Computational Synthesis and Creative Systems. Springer International Publishing, Cham, 2019.

[5] Sergi Jordà. Digital Lutherie Crafting musical computers for new musics ’ performance and improvisation. *Departament de Tecnologia*, 26(3):531, 2005.

[6] Sarah-Indriyati Hardjowirogo. Instrumentality. On the Construction of Instrumental Identity. In *Musical Instruments in the 21st Century Identities, Configurations, Practices*. Springer, Berlin, 2017.

[7] Wayne Siegel. Dancing the Music: Interactive Dance and Music. In Roger T. Dean, editor, *The Oxford Handbook of Computer Music*, chapter 10, pages 191 – 213. Oxford University Press, Oxford, 2009.

[8] Joseph A. Paradiso. Footnotes: Personal reflections on the development of instrumented dance shoes and their musical applications. In *in Quinz, E., ed., Digital Performance, Anomalie, digital*$_arts, pages 34 - -49, 2000$.

[9] A. Di Perna. Tapping into MIDI. *Keyboard Magazine*, page 27, 1988.

[10] Jônatas Manzolli, A. Moroni, and C. Matallo. AtoContAto: new media performance for video and interactive tap shoes music. In *AtoContAto: new media performance for video and interactive tap shoes music.*, page 31, Bristol, 1998.

[11] Ian Hattwick, Joseph Malloch, and Marcelo Wanderley. Forming Shapes to Bodies: Design for Manufacturing in the Prosthetic Instruments. *Proceedings of the International Conference on New Interfaces for Musical Expression*, (August 2016):443–448, 2014.

[12] Kristian Nymoen. MuMYO — Evaluating and Exploring the MYO Armband for Musical Interaction. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 215–218, Bâton Rouge, US, 2015.

[13] Kyle McDonald. DIY Force Sensitive Resistor (FSR): https://www.instructables.com/id/DIY-Force-Sensitive-Resistor-FSR/, 2008.

[14] Andrew P McPherson, Robert H Jack, and Giulio Moro. Action-Sound Latency: Are Our Tools Fast Enough? *Proceedings of the International Conference on New Interfaces for Musical Expression*, 16:20–25, 2016.

[15] Johnty Wang, Axel Mulder, and Marcelo Wanderley. Practical Considerations for MIDI Over Bluetooth Low Energy as a Wireless Interface. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Porto Alegre, Brazil, 2019.

Cognitive Offloading: Can ubimus technologies affect our musicality?

Leandro Costalonga, Marcelo S. Pimenta

NESCoM – Universidade Federal do Espírito Santo
BR 101 Km 60 – 29932-540 São Mateus, ES

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

leandro.costalonga@ufes,br, mpimenta@inf.ufrgs.br

Abstract

An interaction design that lean towards musical traits based on and constrained by our cognitive and biological system could, not only provide a better user experience, but also minimize collateral effects of excessive use of such technology to make music. This paper presents and discuss innate abilities involved in musical activities that - in the authors´ viewpoint - could be considered in design guidelines to computer music technologies, especially those related to ubimus.

1. Introduction

If we consider music as a product of musicality, a social and cultural construct of humankind based on the presence of several cross-cultural similarities, it is very important to investigate how such musicality is affected by the technology we are using and building to provide support to our musical activities. We think everyone is a skilled and sophisticated musical listener, even those who consider themselves to be “unmusical” or do not themselves produce music have an implicit knowledge of the musical forms and styles of their culture (even if they cannot be expressed explicitly). For instance, all individuals have an implicit understanding of the melodic, rhythmic, and harmonic regularities of their cultures' music, in the same way they unconsciously know the structure and rules of our native language [1].

The growing man-machine integration results in a broader understanding of human experience and cognition. The same way that McLuhan has discussed how communication technology (alphabetic writing, the printing press, and the electronic media) affects cognitive organization, we think the use of ubimus technology could also shape the way we think about music or even have any (positive or detrimental) effect on our musicality.

The ability to rely on the external mind might have detrimental consequences to cognition [2] because humans are ‘‘cognitive misers’’, meaning that people tend to eschew costly analytic thought in favour of comparatively effortless intuitive processing [3]. The miserly nature of human cognition lends itself to an overreliance on simple heuristics and mental shortcuts [4, 5]. The evidence suggests that high smartphones’ users are genuinely lower in cognitive ability and have a more intuitive cognitive style. Based on that premise, we are investigating if the use of ubimus technology could also shape the way we think music or has any detrimental effect on our musicality. This paper presents and discuss aspects that could improve the design guidelines to computer music technologies, specially related to ubimus.

For instance, Sparrow & Wegner [6] pointed out that when people expect to have future access to information, they have lower rates of recall of the information itself and enhanced recall for where to access it instead. In a musical context, it is known that short-term memory capacity is crucial to the segmentation strategy used by good sight-readers whenever reading a musical score [7, 8]. Sight reading is especially important in the first stage of the musical performance plan, that is acquiring knowledge of the music and developing preliminary ideas about how it should be performed. According to Gabrielsson [9], it is also in this first stage that the structural analysis reveals the real meaning of the musical information. Such a cognitively demanding task requires a substantial amount of analytical reasoning that, in turn, can be ultimately trusted to our smartphones as demonstrated by Barr et al. [3]. The second stage - the musical performance plan - involves hard work on technical problems in order to establish the spatiomotor pattern required to perform the music. Finally, the third and final stage is a fusion of the two previous stages with trial rehearsals that produces a final version of the performance [9]. The last two stages above mentioned demands executive functioning and anxiety control and yet, once again, the dependence on the smart devices plays a significant disrupting role at performing this task [10, 11].

Despite the evidences, it would be premature to state that the very best technology that has been created in order to provide support to musical activities purpose is, in fact, atrophying our musicality. Nevertheless, it is possible to approach this issue from a different angle, that is, researching about the cognitive and biological traits involved in musical thinking and applying it in the design of new ubimus tech. Doing so, we would lean towards innate and primitive structures related to music making, which is unlikely to change due to behavioral overuse of these technologies.

If musicality can be defined as a natural, spontaneously developing set of traits based on and constrained by our cognitive and biological system, music in all its variety can be defined as a social and cultural construct based on that very musicality [12], as will be discussed in the next section.

2. Musicality: Cognitive and Biological Musical Traits

We all can perceive and enjoy music. Over the years, it has become clear that all humans share a predisposition for music, just like we have for language. To recognize a melody and perceive the beat of music is an example of a trait based on and constrained by our cognitive abilities and their underlying biology (trivial skill for most humans). Even infants are sensitive to such features, which are common across cultures [13, 14]. Other common human traits in musicality reported by Honing [1] are : a) relative pitch (e.g., contour and interval analysis; b) regularity and beat perception; c) tonal encoding of pitch; and d) metrical encoding of rhythm.

Until relatively recently, most scholars were wary of the notion that music cognition could have a biological basis. Music was viewed as a cultural product with no evolutionary history and no biological constraints on its manifestation. This explanation is supported by the belief that music has not been around long enough to have shaped perceptual mechanisms over thousands of generations. Moreover, in contrast to speech, this musical knowledge is acquired relatively slowly and not equally by all individuals of a given nature [15]. Such notions, however, do not explain the presence of music in all cultures and time periods, let alone other species. More recently, studies have indicated that our capacity for music has an intimate relationship with our cognition and underlying biology, which is particularly clear when the focus is on perception rather than production [1, 16, 17].

Comparative research shows that although music itself may be specifically human, some of the fundamental mechanisms that underlie human musicality are shared with other species. For Darwin, music had no survival benefits but offered a means of impressing potential partners, thereby contributing to reproductive success. If so, possibility these cognitive traits are the target of natural selection (bear in mind that cognitive traits are polygenic). Darwin even argued that musical vocalizations preceded language [18].

Impressing potential partners may be a feasible purpose for music, however there are divergent studies on that matter. Other reported purposes for music are: a) promotion and maintenance of group cohesion, working as a glue that enhances cooperation and strengthens feelings of unity [19]; b) ease the burdens of caregiving and promote infant well-being and survival [20]. This view even see such vocalizations as having paved the way not only for language but also for music [21] and c) music is a technology or transformative invention that makes use of existing skills and has consequences for culture and biology [22], While there might be quite some evidence that components of musicality overlap with non-musical cognitive features, this is in itself no evidence against musicality as an evolved biological trait or set of traits. It still has to be demonstrated that the constituent components of musicality, when identified, are indeed domain specific. As in language, musicality could have evolved from existing elements through evolutionary processes, such as natural or sexual selection. Alternatively, based on the converging evidence for music-specific responses along specific neural pathways, it could be that brain networks that support musicality are partly recycled for language, thus predicting more overlap than segregation of cognitive functions.

All in all, consensus is growing that musicality has deep biological foundations, based on accumulating evidence for the involvement of genetic variation [23, 24]. Recent advances in molecular technologies provide an effective way of exploring these biological foundations, such as the association studies of genome aiming to capture the polymorphic content of a large phenotyped population sample.

3. Guidelines: Interaction and UbiMus

So far, based on the referred literature, it has been stablished that: a) there might be cognitive and biological traits related to musical activities; b) some human cognitive skills could be affected by ubiquitous technology, especially connected mobile devices. The question that is now posed is: how to design better ubiquitous technology for musical activities (UbiMus) that makes the most of our innate predisposition to music (musicality) in order to minimize the detrimental cognitive effects of extensive use of such devices?

Historically, digital things made by interaction designers were largely tools intended to be used instrumentally, for solving problems and carrying out tasks, and mostly to be used individually. In this scenario, concepts such as user goals, task flows, usability and utility were (and still are) very valuable. However, it turns out that digital technology in today's society is mostly used for communication (many-to-many), entertainment, and for pleasure. This is where user experience design thrives.

As the name suggests, user experience design is about designing the ideal experience of using a service or product. It is about the way people feel about a product and their pleasure and satisfaction when using it, looking at it, holding it, etc. Every product that is used by someone has a user experience. There are numerous theories, methodologies, and frameworks that help designers to design products focused on the user experience. It is not in the scope of this work to discuss it; however, they all suggest paying close attention to the user's needs and expected behavior (known as User-Centered Design). The user must be in the center of the designing process, they must not only be listened but also be involved. Overall, it is essential to take into account what people are good and bad at, both in a motor and cognitive level. For that reason, Human Computer Interaction (HCI) has been always interconnected with the fields or ergonomics and cognitive sciences [25]. Next, some important aspects of human cognition related to music are presented aiming to guide the development of new computer music technology.

3.1 Establish a reference to be imitated

(True) imitation is innate. It is well-developed in humans being observed in new-borns babies both for fostering learning and for yielding pleasure. There is a distinction between imitation that copies the task structure and hierarchical organization, and imitation that copies movements. True imitation focuses on the goal, in other words, the execution of the action as a function of the goal[26]. In a musical context, can be approached from different viewpoints, such as: imitation skills, musical figures, imitation of symbols, imitation of moving sonic forms (corporeal imitation), and imitation of group behaviour (allelo-imitation).

Playing a musical instrument starts with the imitation of low-level skills and low-level challenges. However, as skills improve, the challenges can rise to a higher level. When skills and challenges are in equilibrium, this gives rise to an optimal experience or pleasure.

Learning to play a musical instrument is, therefore, a typical example of true imitation. It draws on the ability of the student to focus on what is essential in the teacher's example. Even if the instrument is not the same it is still possible to imitate particular behaviours and playing styles because the student has more of a focus on the goals and less of a focus on the precise movements. However, the student's ability to see the movements and gestures of the teacher may be an important component in learning to play a musical instrument. The visual observation of expressive movements may facilitate the mirroring of the teacher's intentions to the student's intentions [27].

The role of mirroring in music education has been confirmed by a brain imaging study [28] . Playing of a musical instrument was used to show that the decomposition into elementary motor components was encoded by the mirror neurons. When the action to be imitated corresponded to an elementary action already present in the mirror neuron system, this act was forwarded to other structures and replicated. In that case, no learning was needed [27].

The conclusion here is that, in order to capitalize on the human innate capability to imitate, the designers of a computational performance tool must take into account ways of facilitating this process of true imitation either by providing key examples as well as awareness of the performer on actions compared to others. Guidance and a reference is needed.

3.2 Building blocks

Barr et al. [3] studies suggest that people who think more intuitively and less analytically when given reasoning problems were more likely to rely on their connected devices, suggesting that people may be prone to look up information that they actually know or could easily learn, but are unwilling to invest the cognitive cost associated with encoding and retrieval. In that sense, a possible approach is to offer building blocks that simplifies a set of complex tasks that can looked closer whenever the user feels prepared to do so. Those abstractions can represent performer's actions, musical structures (i.e. arpeggios), emotional intention, improvisation strategy (with the use of AI), riffs, samples, rhythmic patterns, etc. Not only these blocks should be easily available, searchable, but it should also be suggested based on context and user profiling.

Note, however, that music is still believed to be mostly a matter of the “intuitive” right brain – the avatar of emotion and creativity [15]. If that so, chances are the user will never look into the building blocks since one might choose not to engage in costly elaborative encoding, as they know that knowledge can be procured externally. Therefore, besides being a good approach to manage frustration and anxiety, building blocks strategy might not be ideal, for example, for musical learning software.

3.3 Movements and Gestures

As previously mentioned, people engage with music in a way similar to the way they engage with other people[27] : the process of an appreciation of music - although including also cerebral appreciation and interpretation -is strongly based on mirroring body movement.

In order to sound natural in performance, expressive timing must conform to the principle of human movement [29]. People's tendency to move in synchrony with auditory rhythms is known as **ideomotor principle**: perception of movement will always induce a tendency to perform the same or similar movements [30]. The effect is clearly observable in the tendency to tap along with the beat of the music [27]. The beat is the most natural feature for synchronized movement because it appeals to fundamental biomechanical resonances [31]. In this regard, Knuf et. al. [30] ran a comprehensive study on ideomotor actions and verified that movements did not always occur without awareness, but they did occur without awareness of voluntary control. They have also found clear evidence that people do tend to perform the movements they would like to see (intentional induction) whereas results are less clear with respect to perceptual induction (movements that people actually see). Perceptual induction could only be verified thru noninstrumental effectors: in their experiment, the effect appeared for both head and foot. For hand movements (the instrument effectors), intentional induction is much more pronounced than perceptual.

Corporeal articulation is also related to musical expressiveness and can be seen as indicators of intentionality (studied as seen above in terms of this mirroring process)[27]. In general terms, movement in response to music is often seen as a gestural expression of a particular emotion (sadness, happiness, love, anger) that is assumed to be imitated by the music [32]. Therefore, a fundamental question is: if articulations are a kind of expression, how do they relate to expressiveness in music? [27] Bear in mind that the expression of emotion is only one aspect of corporeal involvement with music since corporeal articulations can also be used to annotate structural features, such as melody, tonality, and percussion events. In summary, our gestures and movements give away our intentions and must be considered when musical activities take place.

3.4 Tempo, timing, regularity

Todd [33] defends the principle that performance, perception of tempo and musical dynamics are based on an internal sense of motion. This principle reflects upon the notion that music performance and perception have their origins in the kinematic and dynamic characteristics of typical motor actions. For example, regularities observed in a sequence of foot movements during walking or running are similar to regularities observed in sequences of beats or note values when a musical performance changes tempo.

A shared assumption from these lines of works is that we experience and make sense of musical phenomena by metaphorically mapping the concepts derived from our bodily experience of the physical world into music. Accordingly, listeners hear the unfolding musical events as shaped by the action of certain musical forces that behave similarly to the forces behind our movements in the physical world such as gravity and inertia[34]. Baily [35] even argues that the performer‘s internal representation of music is in terms of movement, rather than sound.

Honing [36] defends that regularity and beat perception is one of the human’s innate musical traits. Cate et al. [37] goes beyond studding in different species both the ability to recognize the regularity in the auditory stimulus and the ability to adjust the own motor output to the perceived pattern. Although rare, this ability appears to some animals as well.

Human rhythmic abilities obviously did not arise to allow people to synchronize to metronomes but rather to the actions of other humans in groups, known as social synchronization. Thus, by the ecological principle, the concept of mutual entrainment among two or more individuals should be the ability of central interest rather than BPS to a mechanical timekeeper [38]. In the wild (i.e., outside the lab), the mutual entrainment of two or more individuals by such a mechanism obviously cannot occur unless they themselves are capable of producing the evenly paced entraining stimulus of some kind (such as clapping, stomping, or drumming) within the tempo range of its predictive timing mechanism [39].

The most sophisticated form of synchronization involves beat-based predictive timing, where an internal beat is tuned to the frequency and phase of an isochronous time-giver, allowing perfect 0-degree phase alignment. This stimulus makes the very next beat in the sequence predictable, allowing the timing mechanism to align—or latch—its produced behavioral to the stimulus with zero, or even small negative (anticipatory), phase lag, typical of human sensorimotor synchrony [40]. Because of reaction time limitations, it cannot therefore be based on responding to that stimulus event. Instead, it requires a predictive (anticipatory) and cyclical motor timing mechanism that takes an evenly paced stimulus sequence as input. Naturally, reaction times to predictable stimuli are shorter than those to unpredictable ones, hence preparatory cues such as “ready, steady” or a drummer’s count down allow quicker responses.

3.5 Tonal hearing

Humans readily recognize tone sequences that are shifted up or down in log frequency because the pattern of relative pitches is maintained (referred to as interval perception or relative pitch). Humans also have sensitivity to spectral contour of musical signals (like birds) but relative pitch is the preferred mode of listening for humans.

The tonal encoding of pitch also seems to be another innate human ability regarded to music. There is substantial empirical evidence that listeners use this tonal knowledge in music perception automatically. Central to pitch organization is the perception of pitch along musical scales. A musical scale refers to the use of a small subset of pitches in a given musical piece. Scale tones are not equivalent and are organized around a central tone, called the tonic. Usually, a tonal musical piece starts and ends on the tonic. Tonal organization of pitch applies to most types of music, but it does not occur in processing other sound patterns, such as speech. Although the commonly used scales differ somewhat from culture to culture, most musical scales use pitches of unequal spacing that are organized around 5–7 focal pitches and afford the building of pitch hierarchies. The tonal encoding module seems to exploit musical predispositions, as infants show enhanced processing for scales with unequal pitch steps [37].

Innate musical predispositions implies specialized brain structures to deal with music: tonal encoding can be selectively impaired by brain damage; for example, after damage some patients are no longer able to judge melodic closure properly. Recent functional neuroimaging study has identified the rostromedial prefrontal cortex as a likely brain substrate for the 'tonal encoding' module [38]. Musical memory is also organized, at least partially, independent of the hippocampus - the brain structure that is central to memory formation. It is possible that the enormous significance of music throughout all times and in all cultures contributed to the development of an independent memory for music [39].

After all these findings, we need to rethink the specificity of music related brain cells may be not fully exploited by the traditional (ubimus) design decisions.

4. Final Discussion

Music, as we know it, has been done in the same way for centuries. It is true that musical instruments have evolved to provide a better (not ideal) fit to our body, sound quality has been improved, new musical styles appeared while others died, and the musical industry reinvented itself several times over. Nevertheless, the cognitive and biological traits that support all this activity, from producing to enjoying music, has been largely the same. Evolution on this matter comes slow. Fact is, humans excel at doing music this way.

With the popularization of the computer technology, a lot of progress had to be made in field of HCI to allow the computers to be usable by nonspecialized regular people. Currently, computers (in all its modern shapes and forms) are involved in virtually all activities known by human beings, music included. A relevant change of paradigm in HCI was Weiser's [41] vision of ubiquitous technology (UbiComp). [25].The readings of UbiComp concepts and technology by fields such as Music and Arts led to the birth of subfield so called Ubiquitous Music (UbiMus); This field is concerned with ubiquitous systems of human agents and material resources that afford musical activities through creativity support tools. In practical terms, ubimus is music (or musical activities) supported by ubiquitous computing concepts and technology. It relates to concepts such as: portability, mobility, connectivity, availability (including for non-musicians).

More recently, another emerging research field positioned itself at the intersection of Internet of Things, new interfaces for musical expression, ubiquitous music, human–computer interaction, artificial intelligence, and participatory art - it is known as Internet of Musical Thing (IoMusT). From a computer science perspective, IoMusT refers to the networks of computing devices (smart devices) embedded in physical objects (musical things) dedicated to the production and/or reception of musical content. Possible scenarios for IoMusT are: augmented and immersive concert experiences; audience participation; remote rehearsals; music e-learning; and smart studio production [35].

These are very exciting research areas, but the big question is: are we cognitively equipped to make the most of it? Could we thrive in this new way of making music or it will become "yet-another" short-lived cool interface for music making? Is this new technology really paying attention to the way we do things, taking into consideration what we are good and bad at? Clearly, there is an intrinsic exploratory value in these initiatives, and it will certainly lead to unpredictable artistic outcomes, but this is insufficient to answer the questions above. Moreover, is there a price to pay when we rely so heavily in such tech to perform an activity such as music making?

One of the most pertinent questions for the 21st century is how these increasingly intelligent and invasive technologies will affect our minds. The term “extended mind”, has been used in order to refer to the notion that our cognition goes beyond our brains and suggests that individuals may allow their smart devices to do their thinking for or with them [3]. The ability to rely on the external mind might have detrimental consequences to cognition [2].

This paper discussed and presented some innate abilities involved in musical activities that – in the authors´ viewpoint - could be considered in the current designs of digital musical instruments and ubimus technology as a whole. Even we need more experimental work, recent findings of biomusicology and neuroscience indicate musical activities may have idiossincrasies that are not covered by the traditional approaches of interaction design and user experience design.

References

1. Honing, H.: The origins of musicality. The MIT Press, Cambridge, Massachusetts, USA (2018)

2. Carr, N.: The Shallows: what the internet is doing to our brains. W.W. Norton & Company, New York, New York, USA (2010)

3. Barr, N., Pennycook, G., Stolz, J.A., Fugelsang, J.A.: The brain in your pocket: Evidence that Smartphones are used to supplant thinking. Comput. Human Behav. 48, 473–480 (2015)

4. Kahneman, D.: Thinking, fast and slow. (2011)

5. Stanovich, K.E.: The robot’s rebellion : finding meaning in the age of Darwin. University of Chicago Press (2004)

6. Sparrow, B., Liu, J., Wegner, D.M.: Google effects on memory: cognitive consequences of having information at our fingertips. Science. 333, 776–8 (2011)

7. Bean, K.L.: Reading music instead of spelling it. J. Musicol. 1–5 (1939)

8. Gabrielsson, A.: Music performance research at the millennium. Psychol. Music. 31, 221 (2003)

9. Gabrielsson, A.: The performance of music. Deutsch, Diana (1999). The psycho, 501–602 (1999)

10. Clayton, R.B., Leshner, G., Almond, A.: The Extended iSelf: The Impact of iPhone Separation on Cognition, Emotion, and Physiology. J. Comput. Commun. 20, 119–135 (2015)

11. Hartanto, A., Yang, H.: Is the smartphone a smart choice? The effect of smartphone separation on executive functions. Comput. Human Behav. 64, 329–336 (2016)

12. Honing, H., Ploeger, A.: Cognition and the Evolution of Music: Pitfalls and Prospects. Top. Cogn. Sci. 4, 513–524 (2012)

13. Savage, P.E., Brown, S., Sakai, E., Currie, T.E.: Statistical universals reveal the structures and functions of human music. Proc. Natl. Acad. Sci. U. S. A. 112, 8987–92 (2015)

14. Trehub, S.E., Becker, J., Morley, I.: Cross-cultural perspectives on music and musicality. Philos. Trans. R. Soc. Lond. B. Biol. Sci. 370, 20140096 (2015)

15. Repp, B.H.: Some cognitive and perceptual aspects of speech and music. In: Music, Language, Speech and Brain. pp. 257–268. Macmillan Education UK, London (1991)

16. Fitch, W.T.: The biology and evolution of music: A comparative perspective. Cognition. 100, 173–215 (2006)

17. Winkler, I., Háden, G.P., Ladinig, O., Sziller, I., Honing, H.: Newborn infants detect the beat in music. Proc. Natl. Acad. Sci. U. S. A. 106, 2468–71 (2009)

18. Fitch, W.T.: Rhythmic cognition in humans and animals: distinguishing meter and pulse perception. Front. Syst. Neurosci. 7, 68 (2013)

19. Merker, B.H., Madison, G.S., Eckerdal, P.: On the role and origin of isochrony in human rhythmic entrainment. Cortex. 45, 4–17 (2009)

20. Dissanayake, E.: If music is the food of love, what about survival and reproductive success? Music. Sci. 12, 169–195 (2008)

21. Brown, S.: Evolutionary Models of Music: From Sexual Selection to Group Selection. In: Thompson, F.T.S. (ed.) Perspectives in Ethology. pp. 231–281. Springer, Boston, MA, Boston, MA. (2011)

22. Patel, A.D.: Music, language, and the brain. Oxford University Press, New York, NY (2008)

23. Oikkonen, J., Onkamo, P., Järvelä, I., Kanduri, C.: Convergent evidence for the molecular basis of musical traits. Sci. Rep. 6, 39707 (2016)

24. Särkämö, T., Altenmüller, E., Rodríguez-Fornells, A., Peretz, I.: Editorial: Music, Brain, and Rehabilitation: Emerging Therapeutic Applications and Potential Neural Mechanisms. Front. Hum. Neurosci. 10, 103 (2016)

25. Preece, J., Rogers, Y., Sharp, H.: Interaction design : beyond human-computer interaction. Wiley, Chichester, UK. (2015)

26. Byrne, R.W., Russon, A.E.: Learning by imitation: a hierarchical approach. Behav. Brain Sci. 21, 667–84; discussion 684-721 (1998)

27. Leman, M.: Embodied music cognition and mediation technology. MIT Press, Cambridge, Massachusetts (2008)

28. Buccino, G., Binkofski, F., Riggio, L.: The mirror neuron system and action recognition. Brain Lang. 89, 370–376 (2004)

29. Honing, H.: The final ritard: On music, motion, and kinematic models. Comput. Music J. 27, 66–72 (2003)

30. Knuf, L., Aschersleben, G., Prinz, W.: An analysis of ideomotor action. J. Exp. Psychol. Gen. 130, 779–798 (2001)

31. Hallam, S., Cross, I., Thaut, M.: The Oxford handbook of music psychology. Oxford University Press, Oxford, UK (2005)

32. Friberg, A., Sundberg, J.: Does music performance allude to locomotion? A model of final ritardandi derived from measurements of stopping runners. J. Acoust. Soc. Am. 105, 1469–1484 (1999)

33. Todd, N.P.M.: The dynamics of dynamics - a model of musical expression. J. Acoust. Soc. Am. 91, 3540–3550 (1992)

34. Dogantan-Dack, M.: The body behind music: Precedents and prospects. Psychol. Music. 34, 449–464 (2006)

35. Baily, J.: Music structure and human movement. Music. Struct. Cogn. 237–58 (1985)

36. Honing, H.: Without it no music: beat induction as a fundamental musical trait. Ann. N. Y. Acad. Sci. 1252, 85–91 (2012)

37. ten Cate, C., Spierings, M., Hubert, J., Honing, H.: Can Birds Perceive Rhythmic Patterns? A Review and Experiments on a Songbird and a Parrot Species. Front. Psychol. 7, 730 (2016)

38. Fitch, W.T.: Four principles of bio-musicology. Philos. Trans. R. Soc. B Biol. Sci. 370, 20140091–20140091 (2015)

39. Merker, B., Morley, I., Zuidema, W.: Five fundamental constraints on theories of the origins of music. Philos. Trans. R. Soc. B Biol. Sci. 370, 20140095–20140095 (2015)

40. Fraisse, P.: Rhythm and tempo. Psychol. Music. 2, 149–180 (1982)

41. Weiser, M.: The Computer for the 21 st Century. Sci. Am. 265, 94–105 (1991)

Reflection and practice on generative music structuring: the formal problem of ergodicity

Gabriel Montechiari e Silva, Clayton Rosa Mamedes

Department of Arts – Federal University of Paraná
Rua Coronel Dulcídio, 638 - 80420-170 Curitiba, PR

g.monte.silva@gmail.com, claytonmamedes@gmail.com

Abstract

This article approaches the subject of musical form from the standpoint of an algorithmic composition practice. It introduces the problem of ergodicity in music, a formal situation at which music development is perceived as static.

The concepts of *General Periodicity* by Henri Pousseur and *Temporal Gestalt* by James Tenney gave support to a reflection on the nature of the problem, as well as to formulate a twofold structuring procedure based on ideas of continuity and segmentation of the musical course. The devised method was implemented as a computer program to produce entire pieces of music.

1. Introduction

The research presented in this paper is part of a project to develop an automated system of composition that encompasses the production of entire musical pieces. All computer programs we created thus far are parts of a single poietic vision, rather than tools to assist the compositional process. They *are* the compositional process. That being said, on each step of this enterprise a particular issue was brought forth - always based on a subjective evaluation from the composer's point of view. Here we will discuss a formal situation that could be characterized as 'structural silence', and refers to the moment at which a musical piece sounds repetitive.

Curtis Roads pointed out that "one of the great unsolved problems in algorithmic composition is the generation of coherent multilayered structures: mesoform and macroform" [5]. However, the structural problem is not discussed at length in the literature on algorithmic composition. The bibliography on computer music often exhibits a descriptive and catalographic posture. Books like *Computer Music* [2], and the more specific *Composing Music With Computers* [3], and *Algorithmic Composition*[1] [4], provide detailed information about the techniques applied on computer generated music, but they rarely reflect on the musical issues that arise from their implementation. Gerard Nierhaus acknowledged this issue when he observed that "composers publishing material on their algorithmic methods is rather a rare occurrence". He presents some hypothesis to understand why that is, suggesting, for one, that it might be because of a "fear of 'disenchantment' at a piece's creative merit" - a magician never reveals his tricks. Also, algorithmic composition "is mostly only used as a supplementary tool in genuine composition" - this means that by not employing automation throughout the composition process, issues like this do not occur.

Another point made by Nierhaus is that "procedures of algorithmic composition are [...] mainly used in the field of style imitation" [4], meaning that the focus of these works is in assessing how well the techniques perform in replicating an aesthetic. Then, any flaw in macroform or a broader musical context is pointed out as an issue of the *computational* method, rather than *compositional*. In other words, the worries of researchers in the latter kind of works is not that of a composer, but of a musicologist, and their reports reflect that.

The few papers that provide subjective analysis often do so by the use of ambiguous remarks. In this respect, we observed various terms employed to reference the structural problem in algorithmic composition. For instance G. Papadopoulos and G. Wiggins, referring to stochastic methods, stated that "it is difficult to capture higher or more abstract levels of music" [6]. In a similar fashion, Nierhaus pointed out that "in most cases, [a stochastic model] is not able to describe the context sufficiently"[4].

In the same sense, while analyzing the results generated by the program *WolframTones,* Curtis Roads stated that those "start without formal opening and continue mechanically ad infinitum, without taking into account higher forces operating above the sonic surface", and evaluated that "these banal sound patterns [...] operate on only one level of organization; they lack narrative macrostructure, development, sectional organization, and macroform" [5]. The term 'banal' was used almost 60 year prior by R. C. Pinkerton to refer to the lack of structural content in melodies generated by the use of transition tables [1]. In all these cases, the authors refer to the inability of the models in dealing with different hierarchical levels of musical form.

The deficiency we've been talking about is also referred to as a lack of musical *meaning*. Papadopoulos

[1] To make justice, Nierhaus does reflect on algorithmic composition on the conclusion of that book, and also does a remarkable job in his later book *Patterns of Intuition*.

and Wiggins pointed out that some of the techniques used in algorithmic composition "do not make any strong claims about the semantics of the pieces". According to them generative grammars, as an example, "generate a large number of musical strings of questionable quality". The authors stated that "the big disadvantage of most, if not all, the computational models (in varying degrees) is that the music that they produce is meaningless" [5].

Roads suggested that "musical meaning is embedded in layers and encoded in many simultaneous musical parameters or dimensions" [6]. In that sense, we could make a claim in favor of the interpretation that musical meaning emerges from perceiving and responding to the internal relations of a musical piece [7]. And, if "the meaning communicated by a formal system is, essentially, itself [...], a 'tying-together' of ideas and experiences" [8], talking about the lack of meaning would be equivalent to talking about the lack of formal structure.

As pointed out earlier, this structural problem is related to a sense of repetitiveness, but the repetition we refer to is of a special kind. This is not a literal repetition, but an abstract one, that arises from the mechanics behind the ordering of a piece. Silvio Ferraz [9] approached the problem of repetition through referencing some of twentieth century's musical streams. In serial music, specifically, he identified a kind of repetition resulting from "the unifying principle of the material diversity that these works presented". The author said that "if we understand repetition as a notion that extends to whichever element that returns in a given system, [...], we could say that the repetition in serial thought is given as a conceptual repetition" [9]. With the idea of *conceptual repetition* in mind, we can glimpse at why the formal *stasis* problem is constantly lurking behind an algorithmic composition work.

Algorithms are lists of procedures, rules. It has been said that "a rule or law signifies a finite or infinite procedure, always the same, applied to continuous or 'discrete' elements", and "this definition implies the notion of repetition, of recurrence in time" [10]. So, "in order for a rule to exist it must be applicable several times in eternity's space and time", and a periodic property is implied. It would seem, then, that there is no escaping repetitiveness if we are to use algorithmic means of composition.

While there is little written about that in algorithmic composition literature, the way to deal with the two poles of boredom (by excess of variety, or by lack thereof) was widely discussed by twentieth century composers - specially by those on the Darmstadt school. We studied theoretical works of two post-war composers, Henri Pousseur and James Tenney, and through their interpretation on the nature of musical structure we devised and implemented an automated composition method focused on the meso and macro structural level of musical forming.

2. The General Periodicity and Temporal Gestalten

Henri Pousseur, while analyzing *Structures* by Pierre Boulez, questioned why that highly organized serial piece sounded just like one composed by aleatoric processes. To him, this work presents "movements deprived [...] of any individual meaning" [11] (we may remember the terms employed on the criticism of algorithmic processes pointed out at the introduction).

According to Pousseur, a characteristic of total serial music is that, "exception made to the measures and rules themselves, nothing seem to be let to the care of free intent", however, if we focus only on perception - without analytical means - "what we experience is precisely the opposite of this kind of ordering. Precisely in cases in which the most abstract constructs are applied, often we have the impression of being before the workings of some aleatoric principle". In that sense, a piece offers "great resistance to unitary apprehension and distinctive memorization", characteristic to which Pousseur associates the term 'equiprobability', referring to a situation in which "at any moment, anything can happen, at least to our perception" [11].

In *Structures*, the distribution of the material constantly exploits all the variation span. Boulez consciously reached for extreme unpredictability, and he stated "his intention to manifest time's irreversibility, and that means [...] a negation of periodicity, no matter how varied it may be" [11]. This strategy served an ideal of abolishing all musical hierarchy - in this case taken to the last consequence. We may say that in the total serial experiment "the elimination of hierarchies in the realm of simple sound parameters made controlling the contrasts difficult", what resulted "in leveling and consequent complete *stasis* of an even sound surface" [9].

In the music of Anton Webern, Pousseur identified a case of particular equilibrium between the *ethos* of non-polarization and formal clarity. To him, the Austrian composer have succeeded where the next generation have not, and the reason was that Webern "refused to oppose to the course of time", and "definitely recognized its autonomy" [11]. According to Pousseur, the overwhelming material differentiation has a suppressing effect on our perception of time, what is directly linked to *stasis*, since "time is synonym of movement". The author connected this to a wave-like character of the structural/formal issue, by stating that "movement can only be [...] alternated. We always evolve in a finite space; when a transformation develops in a particular direction, it finds sooner or later a concrete boundary, to it unsurpassable" [11]. Then, it would only be possible to ensure a constant impression of movement by means of some level of undulatory development of the musical course. Therefore, music that is fruit of a conscious effort to banish all kind of periodicity would be doomed to formal stagnation.

Behind the idea of *general periodicity* is the notion that every aspect of musical structuring can be interpreted as a wave that modulates a base material. The superposition of complex parametric waves, at different hierarchical levels, establishes a connection between the material and the abstract variation field in which it is set. Pousseur's theory considers the undulatory phenomenon under the perspective of an electronic music composer. In that sense, the author speaks about different *waveforms*, amplitude and frequency modulation, and other concepts that comes from his studio practice. Electronic music prompted the idea of applying compositional procedures to sound generation - the composing of sounds -, but also allowed for the opposite idea. According to Pousseur's theory, musical composition could be generalized as being like audio synthesis. From this perspective, he considered the problem that we've been talking about.

A melodic arch, with a well defined directionality, gives us the impression of half a structural cycle, with a clear wavelength; while the disposition of pitches (or some other parameter) in a constellation-like manner hinders the estimate of an unambiguous wavelength. As Pousseur suggests, our ability of distinguishing parametric periods is linked to the perception of musical structure. Thus, in the latter case, the listener would be unable to sense how the material was structured. It could be said that *structural silence* is similar to white noise and, as our conscience tends to suppress a noise after being exposed to it for some time, we would also alienate ourselves before a formal situation like that. This metaphor contemplates the other extreme of periodicity, with perception suppressing a prolonged single tone.

The solution proposed by Pousseur for keeping a clear formal development lies on the parsimonious use of the variation range in each parameter. According to him, "if we want to produce a wave that is effective in all respects, a wave that establishes *unity*, it will be necessary [...] an economic use of the elementary units that define the variation space"[11]. The composer advocates the conscientious use of periodicity, not its complete suppression, because periodicity and aperiodicity are complementary.

We notice that, in terms of *stasis*, Pousseur refers especially to a chaotic formal situation, and this is not the case for many melodies obtained by generative techniques. However, the reason for *stasis* in an algorithmic music context remains the same as in Pousseur's case: in continuous generation, there will be a point at which all the variation boundaries were already explored - even if, combinatorially, the process keeps producing different material. In that sense, excessive reiteration of an algorithm equates to a drone.

While to Henri Pousseur musical structures behave like *waves*, to James Tenney they behave like *particles*. Tenney's article *Temporal Gestalt Perception in Music* starts with a comparison of two distinct conceptions of time: "for the historian, time is not the undifferentiated 'continuum' of the theoretical physicist, but a hierarchically ordered network of moments" [12]. To Pousseur, musical time was something continuous (as he related structure to continuous functions). To Tenney there is a hierarchical structure that sections time in contextual regions, he stated that "for the musician, a piece of music does not consist merely of an inarticulate stream of elementary sounds, but a hierarchically ordered network of sounds, motives, phrases, passages, sections, movements, etc.". In other words, musical time consists of "time-spans whose perception boundaries are largely determined by the nature of the sounds and sound-configurations occurring within them". These time-spans have the characteristic of being "both internally cohesive and externally segregated from comparable time-spans immediately preceding and following it". Tenney called these time units 'temporal gestalt-units' (TGs).

His article comes from a quest to objectively delineate the borders of the structural groups we form during music perception. Since the book *Meta+Hodos* [13] Tenney has been speculating about adapting the principles of Gestalt psychology (as it was applied to visual stimuli) to the domain of sound. Thus, borrowing the gestalt concepts of *similarity* and *proximity*, the author established a structural unit called *clang*, defined by this statement: "in a collection of sound-elements, those which are *simultaneous* or *contiguous* will tend to form clangs, while relatively greater separations in time will produce segregation, other factors being equal. Those which are *similar* (with respect to values in some parameter) will tend to form clangs, while relative dissimilarity will produce segregation, other factors being equal" [12]. However, on his 1980 article, Tenney looked for creating a formal method of analysis that could be automated and generalized to other structural levels.

The first step Tenney took towards the quantification of his gestalt analysis was a change in the methodological approach. There was a "shift of emphasis from the unifying effects of proximity and similarity to the segregative effects of temporal separation and parametric dissimilarity" [12]. Allied to that, was the strengthening of the notion that any process based on perception should be considered in terms of the passing of time, and not as an 'out of time' analysis. In relation to this last concept, he concluded that the delimitation of *clangs* should be based solely on the identification of the starting point of a new one, as listening proceeds. This point is determined based on a local maxima on the parametric gap between elements. In other words, the point at which a parametric interval is bigger than the interval prior and after it, that's where a new clang should start.

To get to the final version of his model, Tenney needed a way of unifying all the parametric intervals in just one measure of distance. To do that, the author established a metric space. He considered this one of the most important steps to the development of his model: "to treat musical space as a metric space within which all the individual parametric intervals between two points might be integrated into a single measure of distance" [12].

A metric space is a mathematical concept that refers to an ordered pair (M, d), constituted of a set of spatial coordinates M and a metric d (a function to calculate the distance between two points from the set associated with it).

The last step before the generalization of his model could extend it to hierarchical levels beyond the *clang* was deciding on a method to pinpoint the position of gestalt units of a higher level. In summation, Tenney used the centers of mass of each collection of elements.

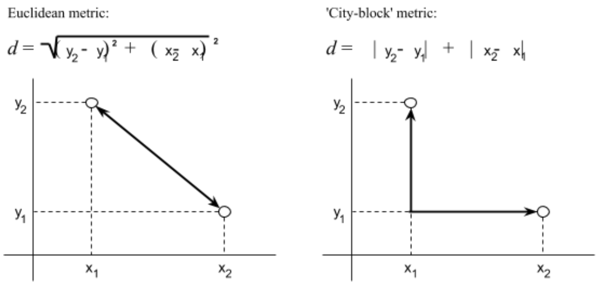


Figure 1: Two examples of metric (adapted from Tenney, 1980, p.213)

The last formulation of his model was: "a new TG at the next higher level will be initiated in perception whenever a TG occurs whose disjunction (with respect to the previous TG at the same hierarchical level) is greater than those immediately preceding and following it".

Based on his theory of temporal gestalten, Tenney noted the same problem tackled by Pousseur: "serial, aleatoric and stochastic compositional methods frequently result in textures which are statistically homogeneous at some fairly low hierarchical level. A typical negative response to this kind of formal situation (which I have elsewhere called 'ergodic') is that, although 'everything is changing, everything remains the same.' [12]. His paper proposed a way of dealing with this situation.

During our research, we often used the term 'ergodicity' to designate the structural problem we've been facing. The word Tenney chose originates from its homonymous applied to the study of dynamic systems. Independently, Iannis Xenakis briefly addressed a definition to the term, that helps us illustrate how it could be applied on a musical context: "the ergodic principle states that the capricious effect of an operation that depends on chance is regularized more and more as the operation is repeated" [10]. In other words, it deals with situations in which a reiterated random process tends to some regularity. In the same way, a reiterated compositional procedure would be subject to this phenomenon. Another reason to highlight the term 'ergodicity' is to explicitly reference Tenney's work, by which he designates a measurable situation. To Tenney, "a piece becomes 'ergodic' (with respect to some parameter) as soon as a hierarchical level is reached [...] at which the mean-intervals between successive TGs are all effectively zero" [12]. If there are not discrepancies in the distance between structures, a new level is not reached and the piece embarks on a static state - an ergodic formal situation.

As we've seen, Henri Pousseur believed the structural problem is caused by prolonged and equanimous exploration of the variation space. James Tenney reached a similar diagnosis, and pointed out that "the more the total available range in some parameter is 'used up' at a given level, the smaller will the average effective difference between TGs at that level, and more quickly will the texture approach 'ergodicity' at the next higher level". Be it through a nexus of complex waves, or through a system of mass points, both authors arrived at the same conclusion in this matter.

3. Our practice

We associated Pousseur and Tenney with two aspects of musical structure: continuity and segmentation. Pousseur’s theoretical work have continuity as a theme, since - even if manifested discreetly - structure was pictured as an abstract wave, a continuous phenomenon. On the other hand, Tenney’s work was about distances as structural determinants, segmentation was his theme. We envisioned a twofold structuring procedure combining ideas from both authors.

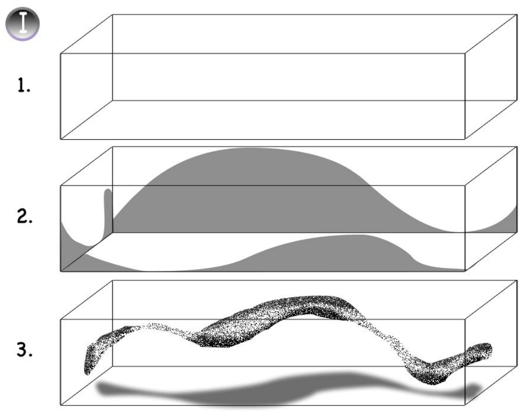


Figure 2: The first step in our compositional model - to institute a continuity in musical space.

The first step in our compositional procedure was focused on establishing continuity. We started by defining a space in which our form would develop (Fig. 2-1). Values of absolute duration, pitch register, and dynamic range, define the fringes of this space. At this stage, two issues remain: which metric to use, and which weights - if any. The metric will determine the way by which we calculate distances, and the weights are the factors by which we redimension the axes. On our implementation, we used euclidean metric and empirically estimated the weights. As these are speculative points on Tenney’s theory, we made sure that different options could be easily tested. There is also a good reason weighting remained a variable: Tenney noticed that each piece analyzed by his method required a different set of weights to optimally correspond to the manually performed analysis. In that sense, the author suggested that the discrepancies in weight values between analysis of different composers were due to differences in the structural role which parameter had on given composer's aesthetic.

Next, parametric profiles are defined (Fig. 2-2), whose function is to modulate the material in a continuous curve in space (Fig. 2-3). This means establishing a conducting line to the musical course. There were profiles to four parameters: global dynamics, pitch transposition, tempo, and a fourth parameter referring to melodic variation. These profiles' shapes are sinusoids that, accordingly to the principle of even exploration of the variation space, have wavelengths corresponding approximately to the duration of the piece - slightly varying on each instantiation. The initial phases of each profile vary in a uniformly random manner.

This simple procedure seemed to prevent the feeling of *stasis*. However, demarcations were lacking between regions of space. Each area of the musical space represents a distinct musical character due to the parameters assigned to each axis, so the continuous transition between them hinders the apprehension of the specifics of form - despite its existence being clearer due to the parsimonious exploration of space. A second procedure was made for sectioning the form, thus emphasizing distinct shapes and the order in which they appear.

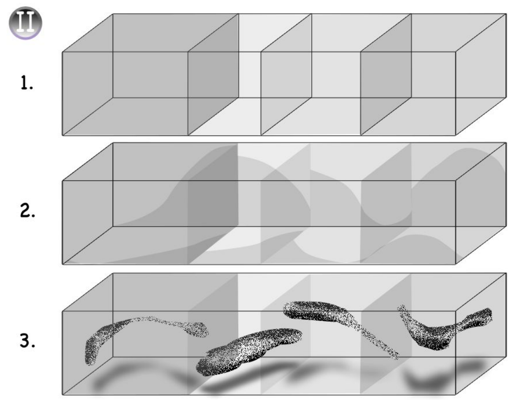


Figure 3: Second step in our compositional model - to break the continuity.

The procedure to introduce structural marks was conceived according to the following principle: they could not derive from material factors - i.e., from repetition of previous material or from introduction of fundamentally distinct material. Our solution was based on the physical phenomenon of refraction. When an object passes through two transparent media, we can observe the boundary between them through the discontinuity of the object's image. Inspired by this, we determined the creation of sections with different refractive indices (Fig. 3-1), and depending on the slope of the parametric profile at the point of junction between sections, the profile's phase is shifted accordingly (Fig. 3-2). The discontinuity thus produced introduces points of reference that help us on the characterization and ordering of distinct moments in a piece (Fig. 3-3).

There is yet a third step, whose function was to promote adjustments on the resulting structure. Its first part consisted of optimizing the distance between structures (Fig. 4-1). For that, it was applied the optimization technique of differential evolution, with a fitness function that judged sets of candidate section displacements, choosing the set with the furthest apart sections. By considering every distance between structures (and not only those between contiguous ones), we avoided a situation in which there is only contrasting structures but no global variation - i.e., we ensured they do not just alternate in space. As a consequence, the exploration of unique zones in space means promoting musical passages with unique characters, what led us to a more effective formal distinction - here, there was a double operation of *unifying* (by continuity) and *differentiating* (by segregation) musical structures.

The melodic material we used came from a generator made on previous works, and have a particular microstructure. The aim of the second adjustment was to emphasize this microstructure (Fig. 4-2). It starts with a structural analysis utilizing Tenney's method, followed by intensity accentuation at the points where a TG starts. This brings up the low-level hierarchy of a section.

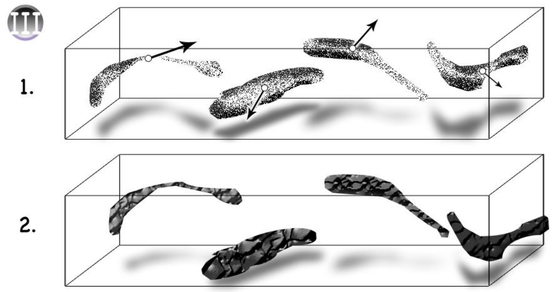


Figure 4: Third step in our compositional method - fine tuning of structure distances, globally [1], and locally [2].

Below we have a picture of five piano-rolls of pieces generated by our program (Fig. 5). Based on them we can make a brief morphological analysis of the process' results. The pitches are on the vertical axis, time is on the horizontal axis, and the intensities are represented by the opacity of each note. Each piece has a particular duration, but some of the images were squashed to fit a uniform column.

The first item (Fig. 5-a) has the clearest section distinctions. There is little register directionality on each section. However, the sections are well spread on this axis. The contrast on the average dynamics contribute to the easy distinction. Tempo[2] is the only attribute that don't seem to contrast between sections, but is a strong

[2] Our pieces don't have metrics, 'tempo' is just a multiplier to the durations of the base material.

conveyor of directionality on this example.

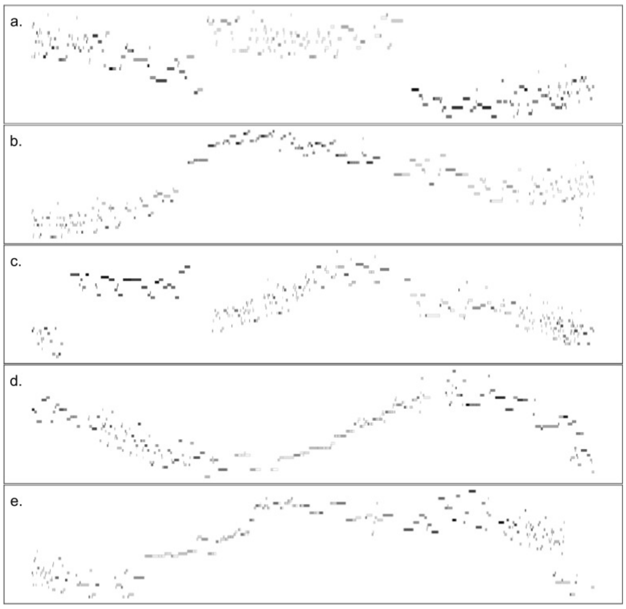


Figure 5: Visual representation of five pieces produced by our program.

An important thing to notice is that some instances don't show a clear sectioning. This happens because, as we mentioned, our method of segregation was inspired by refraction and depending on the angle a parametric profile 'hits' a new section, its phase will be shifted more or less. In item *d* (Fig. 5-d), the sinusoid shape of the profiles is most apparent, and at first glance it seems like there is no sectioning. The reason for this is that the section boundaries are located close to the trough and the crest of the pitch profile, and that causes a smaller shift in that parameter. However, each parameter has a different profile, so they have different displacements between sections, and we can still deduce structural boundaries based on some of them. This is the case of item *b* (Fig. 5-b), where there are also three sections. Between the first two, there is a gap in register, but between the middle and the last section there appears to be a single downward motion on the overall register, and the structural limits would be blurred. However, if we look closely there is a contrast on the average dynamics of both sections, so we can still spot those limits.

4. Conclusion

Our long term objective is the conception of an automated compositional system for producing entire pieces of music. However, until now we were only able to produce musical fragments lacking in high level structure. Given the complexity of the musical phenomenon, there's been moments at which we thought that a complete musical piece couldn't be achieved through automated methods. Now, for the first time since the beginning of our practice, we reached a result that exhibits some structural 'completeness'. Although aesthetically there is much to be done, we concluded that the minimum requirement for complete pieces was met.

At a composer's perspective, the biggest insight our research brought was the realisation that increasingly complicated methods were not the way to overcome the complexity barrier. The formal/structural problem don't require intricate intellectual constructs. It can be greatly simplified by considering it a matter of distributing objects in musical space. Some spatial configurations convey greater complexity than what is needed to describe them. On the other hand, often when these configurations are assembled in a strict logical manner, the complexity is lost in translation to apparent randomness - as it was pointed out by Pousseur, referring to examples in total serialization.

At the end of this research we were left with two guiding principles for structuring music; and one prescription to avoid structural silence, a static formal situation that Tenney called ergodic. The principles relate to the concepts of *continuity* and *separation*, two sides of the same coin. The prescription is to populate the variation space in an economical fashion.

In the field of algorithmic composition there is an attraction towards ideas such as fractal geometry, evolutionary algorithms, cellular automata, neural networks, etc. At times we are tempted to believe that the solutions to our compositional problems will come from technical novelties. Computers make possible to apply meticulous calculations to musical creation, thus allowing us to go further as composers. However, being able to go further means nothing if one doesn't know where to go. Our algorithms are not subject to asymptotic analysis; we don't owe tribute to operational efficiency. Our practice won't benefit from sophisticated computing techniques if there is no solid representational basis supporting it.

The conceptual framework we rely on, and the idea we make of what music is, these are the real origins of the methods we work with as composer-programmers. Computer scientists will focus on algorithmic innovation. To us is left the elaboration of compositional systems capable of manifesting pieces of music in compliance to our inner necessity - even when this necessity is not previously formalized, as in cases where the model and its implementation are built simultaneously. May this not be mistook as a claim for pure intuition, because our *métier* absolutely depends on the formalized method. The point we make is that the method of interest to us a particular one: it is the *creative* method.

"May our imagination rouse our intelligence, and may our intelligence ensure our imagination." [14]

Acknowledgements

This research was supported by the CAPES Higher Education Improvement Coordination.

References

[1] HILLER, L. A.; ISAACSON, L. M. *Experimental*

Music. Nova York: McGraw-Hill, 1959.

[2] DODGE, C.; JERSE, T. A. *Computer Music*: synthesis, composition and performance. 2ª ed. Estados Unidos: Schirmer, 1997.

[3] MIRANDA, Eduardo Reck. *Composing Music with Computers*. Burlington: Focal Press, 2002.

[4] NIERHAUS, Gerard. *Algorithmic Composition*. Alemanha: Springer-Verlag, 2009.

[5] PAPADOPOULOS, G.; WIGGINS, G. *AI Methods for Algorithmic Composition*: A Survey, a Critical View and a Future Prospects. School of Artificial Intelligence, Division of Informatics. Edimburgo: Universidade de Edimburgo, 2000.

[6] ROADS, Curtis. *Composing Electronic Music*. Nova Iorque: Oxford Press, 2015.

[7] MEYER, L. B. *Emotion and Meaning in Music.* Chicago: Chicago Univ. Press, 1956.

[8] COELHO DE SOUZA, R.; FARIA, R. *A Criatividade como Propriedade Emergente na Composição Algorítmica*. SONIC IDEAS. 10. 21-34, 2013.

[9] FERRAZ, Sílvio. *Música e Repetição*: a diferença na composição contemporânea. São Paulo: Educ/FAPESP, 1998.

[10] XENAKIS, Iannis. *Formalized Music*. Revised Edition. Stuyvesant, NY: Pendragon Press, 1992.

[11] POUSSEUR, Henri. *Apoteose de Rameau*. Trad. Flo Menezes e Mauricio Oliveira Santos. Seleção de textos, prefácio e notas críticas de Flo Menezes. São Paulo: Ed. UNESP, 2009.

[12] TENNEY, J.; POLANSKY, L. Temporal Gestalt Perception in Music. Journal of Music Theory, vol. 24, nº 2, pp. 205-241, 1980.

[13] _________. *Meta+Hodus.* Ed. 2. Oakland: Frog Peak Music 1988.

[14] BOULEZ, Pierre. *A música hoje*. Ed. 3. São Paulo: Perspectiva, 2016.

The development of libmosaic-sound: a library for sound design and an extension for the Mosaicode Programming Environment

Luan Luiz Gonçalves[1] **, Flávio Luiz Schiavoni**[1]

[1] Arts Lab in Interfaces, Computers, and Everything Else - ALICE
Computer Science Department – DCOMP
Federal University of São João del-Rei – UFSJ

luanlg.cco@gmail.com, fls@ufsj.edu.br

Abstract. *Music has been influenced by digital technology over the last few decades. With the computer, the musical composition could trespass the use of acoustic instruments demanding to musicians and composers a sort of computer programming skills for the development of musical applications. In order to simplify the development of musical applications, several tools and musical programming languages arose bringing some facilities to lay-musicians on computer programming to use the computer to make music. This work presents the development of a Visual Programming Language (VPL) for audio applications in the Mosaicode programming environment, simplifying sound design and making the synthesis and manipulation of audio more accessible to digital artists. It is also presented the implementation of libmosaic-sound library for the specific domain of Music Computing, which supported the VPL development.*

1 Introduction

Music has been influenced by technology for decades, especially after technological advances, bringing the idea of music and technology together, providing new electronic instruments and new ways of making music. With the computer, musical composition goes beyond the limitations of the artist's body and its acoustic instruments and it started requiring knowledge of computer programming for the development of audio applications and compositions. Since the skills to create a music piece can be totally different from the ability to develop a software, digital artists can find it difficult to start their research and work with digital art due to non-computer programming knowledge.

Fortunately, it is possible to program a computer application using non-textual programming paradigms. Visual Programming Languages (VPLs) allow programmers to develop code using a two-dimensional notation and interacting with the code from a graphical representation [1]. The usage of diagrams to develop applications can make the development easier and allow non-programmers or novice programmers to develop and create software. Furthermore, diagrammatic code abstraction can bring practicality in changing the code, making it suitable for rapid prototyping [2], a feature that can help even experienced programmers. Textual programming paradigms require the use of one-dimensional stream of characters code, demanding the memorization of commands and textual syntax while visual programming languages are more about data flow and abstraction of software functionalities.

Another possibility to further simplify the software development is to use a Domain-Specific (Programming) Language (DSL) [3]. DSLs are at a higher abstraction level than general purpose programming languages because they have the knowledge of the domain embedded in its structure. It makes the process of developing applications within your domain easier and more efficient because DSLs require more knowledge about the domain than programming knowledge [4]. Hence, the potential advantages of DSLs include reduced maintenance costs through re-use of developed resources and increased portability, reliability, optimization and testability [5].

Merging the readiness of VPLs and the higher abstraction of DSLs, we present the Mosaicode , a visual programming environment focused on the development of applications for the specific domain of digital art. The development of an application in the Mosaicode environment, presented in Figure 1, is accomplished by the implementation of a diagram, composed by blocks and connections between them. The schematic of a diagram is used to generate a source code in a specific programming language using a code template for it. The tool also provides resources for creating and editing components (blocks, ports, and code template) to the environment and a set of components is called an extension. Thus, by the creation of new extensions, the tool can be extended to generate code for different programming languages and specific domains – building VLPs for DSLs. Hence, Mosaicode is not restricted to generating applications only for the specific domains of digital art, since it allows the creation of extensions for any other specific domains.

Initially Mosaicode was developed to generate applications to the Computer Vision domain in C/C++ based on the openCV framework. Gradually, new extensions have being developed to attend the digital arts domain bringing together the areas needed to supply the demands of this domain including the processing and synthesis of audio and images, input sensors and controllers, computer vision, computer networks and others [6]. Figure 2 shows the areas involved in the generation of applications for digital art by Mosaicode and highlights the area of Music Computing as the area approached in this work [6].

Apart from the extension to Computer Vision in C/C++ language, Mosaicode also has extensions to generate applications in Javascript/HTML5. These are the current Mosaicode extensions to support the development of applications for specific domains of digital art:

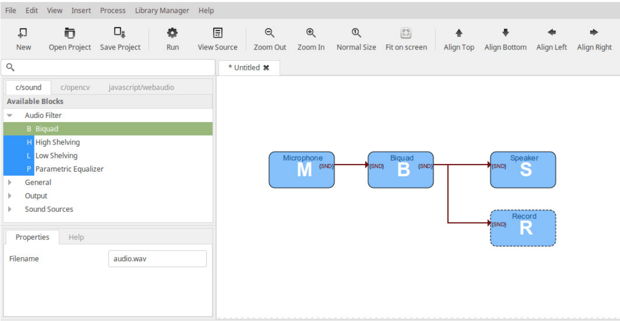


Figure 1: Mosaicode screenshot – the Visual Programming Environment presented in this paper.

(i) **mosaicode-javascript-webaudio**: implements the natives Web Audio API Nodes including the *Script Processor Node* that allows the development of new sound nodes for the Web Audio API. It also implements HTML 5 widgets that compose the generated applications GUI [7]. Further than audio processing and synthesis with Web Audio API, this extension also include other HTML 5 APIs like Web Midi, Gamepad API, Web Socket, WebRTC and others;

(ii) **mosaicode-c-opencv**: implements Computational Vision and Image Processing resources using openCV library for applications generated in the C++ language;

(iii) **mosaicode-c-opengl**: implements Computer Graphics resources using the openGL library based on the C programming language;

(iv) **mosaicode-c-joystick**: based on C language, allows to control applications using joysticks as interface;

(v) **mosaicode-c-gtk**: supports the development of GUI using GTK and C language.

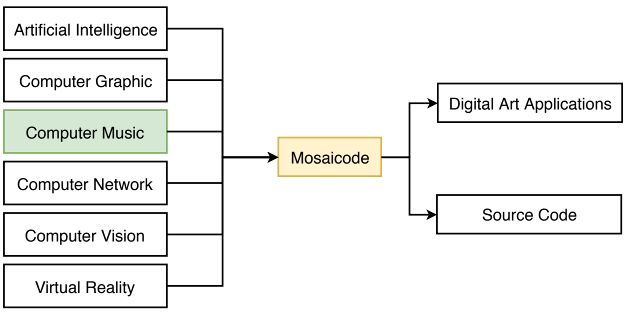


Figure 2: The scheme shows the Computer Science areas involved in the generation of applications for digital art in the Mosaicode.

In order to unleash the development of audio applications in C language, this work presents an extension for the Mosaicode environment focused on the Computer Music domain. This extension in Mosaicode is intended to simplify sound design (manipulation and creation of sounds), making audio synthesis and sound processing more accessible to digital artists. For the language of the generated code was chosen the programming language C, used in the libmosaic-sound library. This library is also the result of this project, developed to assist in the development of the extension, with the aim of facilitating this development by reducing the effort required to implement it.

The libmosaic-sound library was based on the PortAudio API to access the audio input and output system and provide resources that this API does not have implemented on it. To read and write audio files, the libsoundfile API was also used and integrated into our library. So, the user can effortlessly generate applications for the Computer Music domain in C and integrate it with another APIs available for this programming language. The library structure provides this ease of use programming framework and made it easier to implement the blocks in Mosaicode, resulting in the VPL for the Music Computing domain.

2 Related tools

The tools presented below are widely used by digital artists and are considered to be related to this research.

Processing[1] is a programming language and an Integrated Development Environment (IDE) developed by the MIT Media Lab[8]. The programming framework of Processing contains abstractions for various operations with images and drawings and allows rapid prototyping of animations in very few lines of code. The purpose of the tool is to be used for teaching programming and for graphic art development. From programs made in Processing, called sketches, the IDE generates Java code and runs the generated code.

Pure Data[2] or simply PD is a graphical real-time programming environment for audio and video [9] application development. A program in PD is called a*patch* and is done, according to the author himself, through "boxes" connected by "cords". This environment is extensible through plugins, called *externals*, and has several libraries that allow the integration of PD with sensors, Arduino, wiimote, OSC messages, Joysticks and others. PD is an open source project and is widely used by digital artists. The environment engine was even packaged as a library, called libpd [10], which allows one to use PD as a sound engine on other systems like cellphones applications and games.

Max/MSP[3] is also a real-time graphical programming environment for audio and video [11]. Developed by Miller Puckett, the creator of Pure Data, Max is currently maintained and marketed by the Cycling 74 company. Different from the other listed related tools, Max is neither open source or free software.

EyesWeb[4] is a visual programming environment focused on real-time body motion processing and analysis [12]. According to the authors, this information from body motion processing can be used to create and control

[1] Available on `https://processing.org/`
[2] Available on `http://www.puredata.info`
[3] Available at `https://cycling74.com/products/max`
[4] Available on `http://www.infomus.org/`

sounds, music, visual media, effects and external actuators. There is an EyesWeb version, called EyesWeb XMI – for eXtended Multimodal Interaction – intended to improve the ability to process and correlate data streams with a focus on multimodality [13]. Eyesweb is proprietary free and open source with its own license for distribution.

JythonMusic[5] is a free and open source environment based on Python for interactive musical experiences and application development that supports computer-assisted composition. It uses Jython, enabling to work with Processing, Max/MSP, PureData and other environments and languages. It also gives access to Java API and Java based libraries. The user can interact with external devices such as MIDI, create graphical interfaces and also manipulate images [14].

FAUST[6] is a functional programming language for sound synthesis and audio processing. A code developed in FAUST can be translated to a wide range of non-domain specific languages such as C++, C, JAVA, JavaScript, LLVM bit code, and WebAssembly[15].

The present project brings the advantage of Visual programming languages, like Max/MSP and Pure Data and the flexibility of code generation, like FAUST and Processing. All together, this project can be an alternative to these programming languages and programming environments.

3 The extension development

The development of the proposed extension to Mosaicode took three tasks, as depicted in Figure 3, i) a Startup process, ii) the library development and iii) the extension development.

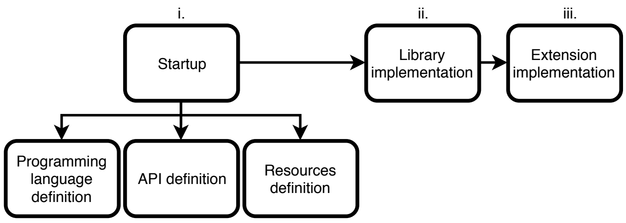


Figure 3: Flowchart of the development methodology of this work splitted into three stages (i, iii and iii).

3.1 The start up process

The first stage of this work, The start up process, was divided into three parts: *1)* choose the programming language for the generated code; *2)* choose the audio API to aid the development and; *3)* define the resources required for a VPL/DSL that enable digital artists to develop audio applications for the Music Computing domain and to work with sound design.

There was a concern to choose a suitable language for the proposed project as well as an API that can simplify the development, bringing resources already implemented, like the access of the audio input and output device, and offering good portability, free software license and allowing the integration with other APIs, like MIDI, OSC and sensors in the future. The process of choosing the language and API was done reading papers and source code of existing tools for audio processing, looking for an efficient API that could bring up the basic resources to develop audio applications.

The choice of the API also influenced the choice of the programming language since the compatibility between both is fundamental to simplify the development of systems. Another concern for implementing audio applications is the efficiency of the programming language. The language chosen should support an efficient audio processing, otherwise the result of the application will not be as expected [16].

Most part of the audio APIs available to audio applications development are developed using the C language [17]. In addition, C is a powerful, flexible and efficient language that has the necessary resources for the development of audio [18],so we chose this programming language for the code generated by Mosaicode. Besides, using the C language could bring interoperability with others extensions present in the environment.

From several APIs available to sound development, the PortAudio API was chosen to simplify the development of the framework in the musical context. Being a cross-platform API, PortAudio allows the implementation of audio streams using the operating system audio APIs, making it possible to write programs for Windows, Linux (OSS/ALSA) and Mac OS X. PortAudio uses the MIT license and can be integrated with PortMidi, a library to work with the MIDI standard [19]. Since PortAudio does not implement access to media files, the *libsoundfile* API was also used to play and record audio files.

After defining the programming language and the audio API, we carried out a survey for a VPL/DSL resources that enable digital artists to develop applications to the Music Computing domain and work with sound design. A list of resources was made based on existing tools, cited in Section 2, and other libraries to develop system to the same domain, like the Gibberish [20] library.

Gibberish has a collection of audio processing classes classified in the following categories: Oscillators, Effects, Filters, Audio Synthesis, Mathematics and Miscellaneous [20]. We have also investigated the native objects of Pure Data and this tool has a list of objects organized in the following categories: General, Time, Mathematics, MIDI and OSC, Miscellaneous, Audio Mathematics, General Audio Manipulation, Audio Oscillators and Tables and Filters of Audio.

By meshing the categories investigated in both tools, the resources were defined to be implemented in Mosaicode in blocks form. For this work we selected some of the resources to be implemented, disregarding resources

[5]Available on `http://jythonmusic.org`
[6]Available on `https://faust.grame.fr/`

that can be implemented by combining others, such as FM synthesis and envelopes. Table 1 presents the resources that have been implemented in the libmosaic-sound library and in the Mosaicode in the blocks form.

Table 1: Resources implemented in libmosaic-sound and in Mosaicode, for generating audio applications.

Categories	Resources/Blocks
Audio Filter	Biquad filter (All-pass, Bandpass, High -pass, Low-pass), High Shelving, Low Shelving and Parametric Equalizer.
Audio Math	Addition, Subtraction, Division and Multiplication.
General	Audio Devices and Channel Shooter Splitter.
Output	Record to audio files e Speaker.
Sound Sources	Oscillators, White Noise, Microphone and Playback audio files.

4 LIBMOSAIC-SOUND Library

With the programming language, API, and resources defined (*in stage i*), the next stage was to implement these resources by developing a library to work with sound design. This library, called libmosaic-sound, had to implement the listed resources looking for an easy way to implement audio applications and requiring less programming effort to complete this task. Existing literature, like the book *DAFX – Digital Audio Effects* [21], aided the implementation of these resources.

We developed a library to make these resources available and easy to use, also thinking about a structure that is adequate for the development of audio applications, making it easier to develop it through code reuse. The library was also designed to not depend on the PortAudio API beyond the access to the audio devices. The PortAudio was used only to list the input and output audio interfaces and to set up the resources of those interfaces. The *libsoundfile* API was used to read a media file and to record audio signals to file.

For each resource, listed in Table 1, an Abstract Data Type (ADT) was implemented following the same pattern, as shown below:

- **input**: input data to be processed. ADTs can have more than one input;
- **output**: processed data. ADTs can have more than one output;
- **framesPerBuffer**: buffer size to be processed in each interaction;
- **process**: function that processes the input data and stores it at the output if the ADT has output;
- **create**: function to create/initialize the ADT.
- **others**: each resource has its properties and values to be stored for processing, so there are variables to store these values.

The implementation also included a *namespace* definition using the *mscsound_* prefix added in library functions, types and definitions to ensure that there were no conflicts with reserved words from other libraries. Another detail of implementation is the audio processing without memory copy, using pointers to reference the same memory address to all processing ADTs. If one needs to process two outputs differently, it is possible to use the ADT called *Channel Shooter Splitter*, which creates a copy of the output in another memory space. That way, there will only be memory copy only spending when it is necessary and clearly defined. The details of how to compile, install, and run the code are described on *README.md* file, available on library's repository at *GitHub*[7].

Source Code at Listing 1 shows the ADT that abstracts the implementation of data capture from a microphone (input device):

Listing 1: ADT Definition *mscsound_mic_t, la* abstracting the microphone implementation.

```
#ifndef MSCSOUND_MIC_H
#define MSCSOUND_MIC_H

typedef struct{
  float *output0;
  int framesPerBuffer;
  void (*process)(void *self, float *);
}mscsound_mic_t;

mscsound_mic_t* mscsound_create_mic(
  int framesPerBuffer);
void mscsound_mic_process();
#endif /* mic.h */
```

The implementation of an application using the libmosaic-sound library depends on some functions that must be defined by the developer and functions that must be called. These functions are described below:

- **mscsound_callback**: a function called to process the input values for every block. This function overrides the PortAudio callback thread copying data read from application's ring buffer to the Portaudio audio output buffer [22]. User must override this library function;
- **mscsound_finished**: function called by the library when the callback function is done. User must also override this function;
- **mscsound_initialize**: function that user must call to initialize the audio application;
- **mscsound_terminate**: function that the user must call to end up the audio application. This function finish the library cleaning up memory allocation.

As an example of the libmosaic-sound library usage, Figure 4 presents the running flow of a code that capture the audio with a microphone, store the audio in an audio file and send the audio the computer speaker.

The ADTs also have some code patterns in the library definition. Some code parts called *declaration*, *execution*, *setup* and *connections* have been defined so one

[7] Available at `https://github.com/Mosaicode/libmosaic-sound/blob/master/README.md`

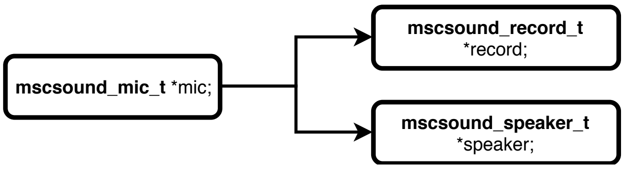


Figure 4: The running flow of a simple audio application developed with the ADTs of the libmosaic-sound library.

can use these code parts to define the implementation of the audio application, being that:

- **declaration**: code part to declare the ADTs used in the code.
- **execution**: code part to define the call order of the *process* functions of each ADT declared in the code part *declaration*. This part must be included within the Mosaicode callback function;
- **setup**: code part to initialize ADTs variables, defining their values and calling their respective *create* functions;
- **connections**: part of the code to define the connections between the ADTs. These connections defines the audio processing chain associating the output of a ADT to the input of another ADT.

For the identification of the code parts, several examples have been created using all the library ADTs. Looking at these examples, it was possible to identify the characteristics of each code part listed above. The implementation code of the libmosaic-sound library and the examples are available on GitHub[8].

5 MOSAICODE-C-SOUND Extension

The last stage (*stage iii*) consisted in the implementation of the extension to develop audio application within the Mosaicode programming environment and using the library previously developed to complete this task.

To create the extension, properties such as name, programming language, description, command to compile, code parts and code template implementation have been defined. The code template informs the code generator of the Mosaicode how to generate source code. By setting the code template, the Mosaicode generator can interpret the diagram and generate the desired source code. Thus, the first step of this stage was to observe in the library and examples developed in stage *ii* the code parts that are common in every example, independently of the implementation, and the unusual parts that are different in every code example.

The code parts that are generated from the diagram are those cited in the development of the libmosaic-sound library in Section 4 – *declaration*, *execution*, *setup* e *connections*. The remaining code will always be the same in all implementations, so it is fixed in the code template.

[8] Available at `https://github.com/Mosaicode/libmosaic-sound`

The second step was to create the input/output port types for the blocks connections, which in this case was just one type, the sound-type port. The connection code has also been defined, establishing how an output block port must be connected to an input block port. The Mosaicode automatically generates these connections by interpreting the block diagram. With the code template and the port created, the last step was the implementation of the blocks for the Mosaicode.

Each developed block contains the code abstraction of a resource defined in stage *i*. This strategy allows a reuse of code by using the library developed in stage *ii*. The Mosaicode blocks can have dynamic and static properties. Dynamic properties can be changed at run-time using the block input ports. Static properties can be changed at programming time, before generating the source code.

The implementation code of the mosaicode-c-sound extension – blocks, ports and code template – are available on GitHub[9].

6 Results

This work resulted in a library for audio application development packed as an extension to the Mosaicode programming environment defining a Visual Programming Language to musical applications development. With this VPL, we simplified application development for Computer Music domain, allowing to generate audio applications and work with sound design by dragging and connecting blocks. We hope it can increasing the facility of digital artists to work with audio applications development.

The developed VPL brings all the resources offered by the libmosaic-sound library, including simple waveform sound sources, enabling the implementation of audio synthesis, sound effects and envelopes to the generation of more complex sounds. It is possible to implement classic synthesizing examples like AM, FM, additive and subtractive synthesizers and implement other techniques of Computer Music, without worrying about code syntax and commands, just dragging and connecting blocks. The user also has the option to obtain the source code of the application defined by the diagram, having complete freedom to modify, study, distribute and use this code.

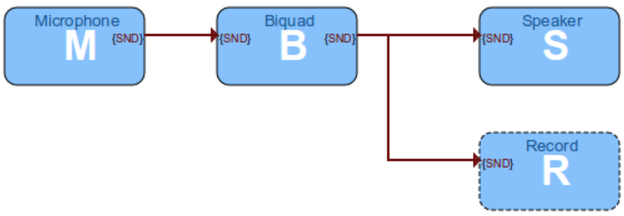


Figure 5: Example of a Mosaicode diagram using the mosaicode-c-sound extension.

Figure 5 shows a diagram as an example of using the extension developed in this work. In this example we

[9] Available at `https://github.com/Mosaicode/mosaicode-c-sound`

apply the lowpass filter (Biquad) to an audio signal captured by a microphone. The filter output is directed to the speaker and recorded in an audio file. The code generated from the diagram of the Figure 5 is shown next. Another examples is available in the extension repository, already available in this document in the Section 5.

Listing 2: Code generated from the diagram in Figure 5.

```
#include <mosaic-sound.h>
#include <portaudio.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NUM_SECONDS 12
#define SAMPLE_RATE 44100
#define FRAMES_PER_BUFFER 256

/* Declaration part */
mscsound_mic_t *block_1;
mscsound_biquad_t *block_2;
mscsound_record_t *block_3;
mscsound_speaker_t *block_4;

static int mscsound_callback(
  const void *inputBuffer,
  void *outputBuffer,
  unsigned long framesPerBuffer,
  const PaStreamCallbackTimeInfo
    *timeInfo,
  PaStreamCallbackFlags statusFlags,
  void *userData) {

    float *in = (float *)
      inputBuffer;
    float *out = (float *)
      outputBuffer;

    (void)timeInfo; /* Prevent unused
      variable warnings. */
    (void)statusFlags;
    (void)userData;

    /* Execution and Connection
parts */
    block_1->process(block_1, in);

    block_2->input = block_1->output;
    block_2->process(block_2);

    block_3->input = block_2->output;
    block_3->process(block_3);

    block_4->input = block_2->output;
    block_4->process(block_4, out);

    return paContinue;
}

static voimscsound_finished(
  void *data) {
    printf("Stream Completed!\n"); }

int main(int argc, char *argv[]) {

  /* Setup part */
  block_1 = mscsound_create_mic(
    FRAMES_PER_BUFFER);
  block_2 = mscsound_create_biquad(
    1, 2, FRAMES_PER_BUFFER);
  block_2->sampleRate = SAMPLE_RATE;
  block_2->cutOff = 3000.0;
  block_2->slope = 0.1;
  block_3 = mscsound_create_record(
    "examples/record_mic_lowpass.wav",
    FRAMES_PER_BUFFER, 44100);
  block_4 = mscsound_create_speaker(
    FRAMES_PER_BUFFER);

  void *stream = mscsound_initialize(
    SAMPLE_RATE, FRAMES_PER_BUFFER);

  printf("Recording until
    the Enter key is pressed.\n");
  getchar();

  mscsound_terminate(stream);
  return 0;
}
```

Comparing the generated code with implementations using PortAudio API, we can notice that the audio structure was maintained. The difference is that function calls are made instead of implementing the abstracted code in these functions.

7 Conclusion

This work proposes the development of an extension for audio application development within the Mosaicode visual programming environment. This development allows the generation of source code from diagrams composed of blocks and connections, making the sound design more accessible to digital artists.

In the first stage of this project, a study has been done to define the appropriate programming language and audio APIs to complete the proposed task. There was also a need to define the necessary resources for a DSL/VPL that would supply the needs of digital artists in the development of applications for the Computer Music domain. In addition, research of the Related tools, like Pure Data, and the Gibberish library helped to define these resources.

In the second stage we discussed the development of the libmosaic-sound library, which supported the implementation of the mosaicode-c-sound extension for the Mosaicodeand allows the development of audio applications in an easier way. The library structure is analogue the manipulation of Mosaicode blocks and connections, as if each ADT is a block and each assignment between output and input was a connection. This structure has also drastically reduced the number of lines a user needs to write developing an audio application compared to the direct use of the PortAudio API. It happens mainly because this API provides only the manipulation of input and output interfaces, requiring the user to implement the processing of the data read/written by the interfaces to generate the applications.

In the third stage we discussed the development of the mosaicode-c-sound extension to work with sound design in the Mosaicode. This extension was based on the libmosaic-sound library, in which each block uses a resources developed and present on the library. In this way, only library function calls are made, making it easier to implement the blocks and generating a smaller source code. This implementation resulted in a VPL for the Computer Music domain and, because it was developed in Mosaicode, allows the generation of the source code that can be studied and modified. In addition, it contributes to Mosaicode with one more extension.

For implementation of the code template in Mosaicode, first, we created several examples of codes using the libmosaic-sound library. These examples have been studied in order to understand each code part and define which parts are fixed in the code template and which parts are generated by the extension blocks.

This project also contributed to the development of Mosaicode, which has undergone code refactoring in order to improve its structure and simplify its maintenance and extension.

7.1 Future works

We intended to review the list of resources in order to expand the library and the extension for audio application. It is also intended to link this project to other projects of the Mosaicode development team. There are several works in progress implementing extensions to Digital Image Processing, Computer Vision, Artificial Intelligence, Computer Networking and Virtual Reality domains. The intention is to connect all these extensions in the environment, offering resources to generate more complex applications for the specific domains of digital art.

8 Acknowledgments

Authors would like to thanks to all ALICE members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq and FAPEMIG.

References

[1] Paul E. Haeberli. Conman: A visual programming language for interactive graphics. *SIGGRAPH Comput. Graph.*, 1988.

[2] Daniel D Hils. Visual languages and computing survey: Data flow visual programming languages. *Journal of Visual Languages & Computing*, 1992.

[3] R. C. Gronback. *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*. Addison-Wesley, 2009.

[4] Marjan Mernik, Jan Heering, and Anthony M Sloane. When and how to develop domain-specific languages. *ACM computing surveys (CSUR)*, 2005.

[5] Arie Van Deursen and Paul Klint. Domain-specific language design requires feature descriptions. *CIT. Journal of computing and information technology*, 2002.

[6] Flávio Luiz Schiavoni and Luan Luiz Gonçalves. From virtual reality to digital arts with mosaicode. In *2017 19th Symposium on Virtual and Augmented Reality (SVR)*, pages 200–206, Curitiba - PR - Brazil, Nov 2017.

[7] Flávio Luiz Schiavoni, Luan Luiz Gonçalves, and André Lucas Nascimento Gomes. Web audio application development with mosaicode. In *Proceedings of the 16th Brazilian Symposium on Computer Music*, pages 107–114, São Paulo - SP - Brazil, 2017.

[8] Casey Reas and Ben Fry. *Processing: a programming handbook for visual designers and artists*. Mit Press, 2007.

[9] Miller S Puckette et al. Pure data. In *ICMC*, 1997.

[10] Peter Brinkmann, Peter Kirn, Richard Lawler, Chris McCormick, Martin Roth, and Hans-Christoph Steiner. Embedding pure data with libpd. In *Proceedings of the Pure Data Convention*. Citeseer, 2011.

[11] Matthew Wright, Richard Dudas, Sami Khoury, Raymond Wang, and David Zicarelli. Supporting the sound description interchange format in the max/msp environment. In *ICMC*, 1999.

[12] Antonio Camurri, Shuji Hashimoto, Matteo Ricchetti, Andrea Ricci, Kenji Suzuki, Riccardo Trocca, and Gualtiero Volpe. Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal*, 2000.

[13] Antonio Camurri, Paolo Coletta, Giovanna Varni, and Simone Ghisio. Developing multimodal interactive systems with eyesweb xmi. In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, NIME '07, pages 305–308, New York, NY, USA, 2007. ACM.

[14] Bill Manaris, Blake Stevens, and Andrew R Brown. Jythonmusic: An environment for teaching algorithmic music composition, dynamic coding and musical performativity. *Journal of Music, Technology & Education*, 9(1):33–56, 2016.

[15] Yann Orlarey, Dominique Fober, and Stéphane Letz. Faust: an efficient functional approach to dsp programming. *New Computational Paradigms for Computer Music*, 290:14, 2009.

[16] Z. Jiang, R.W. Allred, and J.R. Hochschild. Multi-rate digital filter for audio sample-rate conversion, 2002.

[17] Flávio Luiz Schiavoni, Antonio José Homsi Goulart, and Marcelo Queiroz. Apis para o desenvolvimento de aplicações de áudio. *Seminário Música Ciência Tecnologia*, 2012.

[18] Lars Ole Andersen. *Program analysis and specialization for the C programming language*. PhD thesis, University of Cophenhagen, 1994.

[19] Mark Nelson and Belinda Thom. A survey of real-time midi performance. In *Proceedings of the 2004 conference on New interfaces for musical expression*, pages 35–38. National University of Singapore, 2004.

[20] Charles Roberts, Graham Wakefield, and Matthew Wright. The web browser as synthesizer and interface. In *NIME*. Citeseer, 2013.

[21] Udo Zolzer. *DAFX: Digital Audio Effects*. Wiley Publishing, 2nd edition, 2011.

[22] Marc H Sosnick and William T Hsu. Implementing a finite difference-based real-time sound synthesizer using gpus. In *NIME*, pages 264–267, 2011.

Combining Effects in a Music Programming Language based on Patterns

André Rauber Du Bois[1]*, Rodrigo Geraldo Ribeiro[2]

[1]Programa de Pós-Graduação em Computação
Universidade Federal de Pelotas, Pelotas-RS, Brazil

[2]Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Ouro Preto, Ouro Preto - MG, Brazil

dubois@inf.ufpel.edu.br, rodrigo@decsi.ufop.br

Abstract. *HMusic is a domain specific language based on music patterns that can be used to write music and live coding. The main abstractions provided by the language are patterns and tracks. Code written in HMusic looks like patterns and multi-tracks available in music sequencers, drum machines and DAWs. HMusic provides primitives to design and combine patterns generating new patterns. The objective of this paper is to extend the original design of HMusic to allow effects on tracks. We describe new abstractions to add effects on individual tracks and in groups of tracks, and how they influence the combinators for track composition and multiplication. HMusic allows the live coding of music and, as it is embedded in the Haskell functional programming language, programmers can write functions to manipulate effects on the fly. The current implementation of the language is compiled into Sonic Pi [1], and we describe how the compiler's back-end was modified to support the new abstractions for effects. HMusic can be and can be downloaded from [2].*

1 Introduction

Computer music is usually associated with the use of software applications to create music, but on the other hand, there is a growing interest in programming languages that let artists write software as an expression of art. There are a number of programming languages that allow artists to write music, e.g., CSound [3], Max [4, 5], Pure Data [6], Supercollider [7], Chuck [8], FAUST [9], to name a few. Besides writing songs, all these languages also allow the live coding of music. Live coding is the idea of writing programs that represent music while these programs are still running, and changes in the program affect the music being played without breaks in the output [10].

HMusic [11] is a Domain Specific language for music programming and live coding. HMusic is based on the abstraction of patterns and tracks where the code looks very similar to the grids available in sequencers, drum machines and DAWs. The difference is that these abstractions have an inductive definition, hence programmers can write functions that manipulate these tracks in real time. As the DSL is embedded in Haskell, it is possible to use all the power of functional programming in our benefit to define new abstractions over patterns of songs.

This paper discusses new abstractions for HMusic to deal with effects. More precisely, the contributions of this paper are as follows:

- We extend the abstractions of HMusic to incorporate effects. Basically, two new types of tracks are added: a track that takes a list of effects that are applied in order to the track's pattern, and a master track that applies a set of effects to a multi-track (Section 3.1)
- HMusic provides two operators for combining multi-tracks, a sum operator that takes two multi-tracks and generates a new track that plays the two multi-tracks one after the other, and a multiplication operator that takes an integer *n* and a multi-track *t* and generates a track that is *n* times *t*. We extend the behaviour of these operations to deal with effects and explain the semantics of track composition in the presence of effects (Section 3.3)
- We show how the new abstractions for effects can be used during a live coding session (Section 4)
- We describe how the new abstractions presented in this paper can be compiled into Sonic Pi code (Section 4)

To understand the paper the reader needs no previous knowledge of Haskell, although some knowledge of functional programming and recursive definitions would help. We try to introduce the concepts and syntax of Haskell needed to understand the paper as we go along.

The paper is organized as follows. First we describe HMusic and the main constructors for pattern (Section 2.1) and track (Section 2.2) design and their basic operations. Next, the extensions for effects are explained (Section 3.1). In Section 3.3, we examine the semantics of track composition, i.e., combining different multi-tracks to form a new track, in the presence of effects. Live coding with effects is explained in Section 4. The compilation of HMusic with effects into Sonic Pi is described in Section 5. Finally, related work, conclusions and future work are discussed.

2 HMusic

2.1 HMusic Patterns

HMusic is an algebra (i.e., a set and the respective functions on this set) for designing music patterns. The set of

*Supported by CAPES.

all music patterns can be described inductively as an algebraic data type in Haskell:

```
data MPattern = X | O
      | MPattern :|  MPattern
```

The word `data` creates a new data type, in this case, `MPattern`. This definition says that a pattern can be either playing a sample (`X`), a rest (`O`), or a sequential composition of patterns using the operator (`:|`), that takes as arguments two music patterns and returns a new pattern.

As an example, we can define two 4/4 drum patterns, one with a hit in the 1st beat called `kick` and another that hits in the 3rd called `snare`.

```
kick :: MPattern
kick =  X :| O :| O :| O

snare :: MPattern
snare = O :| O :| X :| O
```

The symbol (`::`) is used for type definition in Haskell, and can be read as *has type*, e.g. `kick` has type `MPattern`.

As `MPattern` is a recursive data type, it is possible to write recursive Haskell functions that operate on patterns. For example, usually a certain pattern is repeated many times in a song, and a repeat operator (`.*`) for patterns can be defined as follows:

```
(.*) :: Int -> MPattern
            -> MPattern
1 .* p = p
n .* p = p :| (n-1) .* p
```

The repeat operator takes as arguments an integer `n` and a pattern `p`, and returns a pattern that is a composition of `n` times the pattern `p`. As can be seen in the previous example, the composition operator (`:|`) can combine drum patterns of any size and shape, e.g.:

```
hihatVerse :: MPattern
hihatVerse = 8 .* (X :| O :| X :| O)

hihatChorus :: MPattern
hihatChorus = 4 .* (X :| X :| X :| X)

hihatSong :: MPattern
hihatSong = hihatVerse :|
            hihatChorus :|
            hihatVerse :|
            hihatChorus
```

or simply:

```
hihatSong :: MPattern
hihatSong = 2 .* (hihatVerse :|
                  hihatChorus)
```

In order to make any sound, a pattern must be associated to an instrument hence generating a `Track`, as explained in the next Section.

2.2 HMusic Tracks

A track is the HMusic abstraction that associates an instrument to a pattern. The `Track` data type is also defined as an algebraic type in Haskell:

```
data Track =
   MakeTrack Instrument MPattern
   | Track :|| Track

type Instrument = String
```

A simple track can be created with the `MakeTrack` constructor, which associates an `Instrument` to a `MPattern`. A `Track` can also be the *parallel* composition of two tracks, which can be obtained with the `:||` operator. `Instrument` is a type synonym for `Strings`. An instrument can be any audio file accessible by the Sonic Pi environment (see Section 5).

Now, we can use the previously defined patterns `kick` and `snare` to create tracks:

```
kickTrack :: Track
kickTrack = MakeTrack "BassDrum" kick

snareTrack :: Track
snareTrack =
   MakeTrack "AcousticSnare" snare
```

and also multi-tracks:

```
rockMTrack :: Track
rockMTrack =
 kickTrack :||
 snareTrack :||
 MakeTrack "ClosedHiHat" (X:|X:|X:|X) :||
 MakeTrack "GuitarSample" X
```

3 Effects in HMusic

In this paper, the abstractions of HMusic are extended to incorporate effects. The new abstractions allow to add effects in individual tracks (Section 3.1) and in a group of tracks (Section 3.2). The use of effects in live coding is discussed in Section 4.

3.1 Effects on Tracks

To incorporate effects on tracks, the `MTrack` data type was extend with a new type of track:

```
data MTrack = (...)
  | MakeTrackE Instrument [Effect] MPattern
```

Besides `Instruments` these tracks can take as argument a list of effects that are applied in order. In the current implementation, effects available in Sonic Pi can be loaded in tracks (see Section 5), like changing the rate of samples, reverb, amp, etc:

```
data Effect = Reverb Float | Amp Float
     | Attack Float | Rate Float
     | Sustain Float | (...)
```

For example, we can now write a drum multi-track which adds a bit of reverb on the snare:

```
drums :: MTrack
drums =
 MakeTrackE "snare" [Reverb 0.3] snare :||
 MakeTrack "kick" kick :||
 MakeTrack "hihat" (X:|X:|X:|X)
```

As the MTrack data type has an inductive definition, we can write recursive functions that manipulate effects and tracks (e.g., add, remove, modify) while music is being played, as described in Section 4.

3.2 Effects on Groups of Tracks

HMusic was also extend to support the addition of effects in a group of tracks:

```
data MTrack = (...)
  | Master  [Effect] MTrack
```

The Master track takes two arguments, a list of effects and an MTrack, which is possibly a multi-track, and ads these effects in order to the MTrack.

For example, we can now add effects to the whole drums track defined previously:

```
drumsE :: MTrack
drumsE = Master [Amp 0.2, Sustain 0.4] drums
```

Since the snare track already has a reverb applied to it, when the Master track is added, the effects applied to the snare are now Amp, followed by Sustain, followed by Reverb.

3.3 Effects and Track Composition

The :|| operator, discussed previously, allows the parallel composition of Tracks, i.e., adding an extra track to a multi-track song. But what if we want to combine tracks in sequence, e.g., we have different multi-tracks for the introduction, verse and chorus, and want to combine them in sequence to form a complete song?

If we reason that different multi-tracks represent different parts of a song, when we combine two multi-tracks, we want the second multi-track to start playing only when the first one finished. Hence, when combining a multi-track, its size is always the size of its largest pattern. Thus, when combining tracks we assume that smaller patterns have rest beats at their end, meaning that all patterns are assumed to have the size of the largest pattern in a multi-track. We can define this concepts formally with the following recursive functions:

```
lengthMP :: MPattern -> Int
lengthMP (x:|y) = lengthMP x +
                  lengthMP y
lengthMP _ = 1


lengthTrack :: Track -> Int
lengthTrack (MakeTrack _ dp) =
  lengthMP dp
lengthTrack (MakeTrackE _ _ dp) =
  lengthMP dp
lengthTrack (Master _ t) = lengthTrack t
lengthTrack (t1 :|| t2) =
   max (lengthTrack t1) (lengthTrack t2)
```

Where lengthMP recursively calculates the size of a pattern, and lenghtTrack finds out the size of the largest pattern in a track, i.e., the size of the track.

HMusic provides two constructs for composing tracks in sequence, a repetition operator |* and a sequencing operator |+. The repetition operator is similar to .* but operates on all patterns of a muti-track:

```
|* :: Int -> Track -> Track
```

It takes an integer n and a multi-track t and repeats all patterns in all tracks n times, adding the needed rest beats at the end of smaller tracks.

An operator for combining two multi-tracks t1 and t2, generating a new multi-track is also provided:

```
|+ :: Track -> Track -> Track
```

When combining two multi-tracks, tracks that use the same instruments and effects are merged. The semantics of composing two multi-tracks t1 and t2, i.e., t1 |+ t2 is as follows:

- First we add rest beats to the end of each track in t1 that has matching instruments and effects with tracks in t2, so that all those tracks have the same size as the largest pattern in t1
- Then, for all patterns p1 in t1 and p2 in t2 that have the same instrument i and effects e, we generate a new track MakeTrack i (p1:|p2) for simple tracks, and MakeTrackE i e (p1 :| p2) for tracks with effects. Master tracks are only merged if their internal tracks have the same instruments and effects. Internal tracks of a master track are merged using the rules stated above.
- Finally, we add a pattern of rests the size of t1, to the beginning of all tracks in t2 that were not composed with tracks in t1 in the previous step

Hence the size of the composition of two tracks t1 and t2 is sum of the size of the largest pattern in t1 with the largest pattern in t2.

In Figure 1 some examples of track combination are presented, where track te1te2 is equivalent to t1 |+ te2, and te3te4 is equivalent to te3 |+ te4, and te3twice is equivalent to 2 |* te3.

4 Effects and Live Coding

4.1 Live Coding with HMusic

HMusic provides a set of primitives for playing tracks and live coding. These primitives allow programmers to play songs written in HMusic, loop tracks, and to modify tracks on the fly, i.e., while they are being played. These primitives can be seen in Figure 2.

The first primitive, play, takes two arguments: a Float, which is the BPM (Beats per Minute) of the song and a track, and simply plays this track in the BPM provided. The loop function also takes the same arguments

```
te1 =
  MakeTrack "bassDrum"                          (X :| O :| O)
  :|| MakeTrackE "snare"   [Amp 0.5]            (O :| O :| X)
  :|| MakeTrackE "cymbal"  [Reverb 0.3]         (X :| X :| X :| X)

te2 =  MakeTrack "bassDrum"           (X :| O :| O :| O)
  :|| MakeTrackE "snare" [Amp 0.5]    (O :| O :| X :| O)
  :|| MakeTrack "HiHat"               (X :| O :| X )
  :|| MakeTrack "GuitarSample"         X


te3 = Master [Reverb 1.0] te1
      :|| MakeTrack "Cowbell"        (X :| O :| X )

te4 = Master [Reverb 1.0] te1
   :|| MakeTrack "GuitarSample"       X

te1te2 =  MakeTrack "bassDrum"             (X :| O :| O :| O :| X :| O :| O :| O)
  :|| MakeTrackE "snare" [Amp 0.5]         (O :| O :| X :| O :| O :| O :| X :| O)
  :|| MakeTrackE "cymbal" [Reverb 0.3]     (X :| X :| X :| X)
  :|| MakeTrack "HiHat"                    (O :| O :| O :| O :| X :| O :| X)
  :|| MakeTrack "GuitarSample"             (O :| O :| O :| O :| X)

te3te4 = Master [Reverb 1.0]
         (MakeTrack "bassDrum              (X :| O :| O :| O :| X)
       :|| MakeTrackE "snare" [Amp 0.5]    (O :| O :| X :| O :| O :| O :| X)
       :|| MakeTrackE "cymbal" [Reverb 0.3] (X :| X :| X :| X :| X :| X :| X :| X))
 :|| MakeTrack "GuitarSample"              (O :| O :| O :| O :| X)
 :|| MakeTrack "Cowbell"                   (X :| O :| X)

te3twice =
  Master [Reverb 1.0]
     (MakeTrack "bassDrum"                (X :| O :| O :| O :| X :| O :| O :| O)
      :|| MakeTrackE "snare" [Amp 0.5]    (O :| O :| X :| O :| O :| O :| X :| O)
      :|| MakeTrackE "cymbal" [Reverb 0.3] (X :| X :| X :| X :| X :| X :| X :| X))
 :|| MakeTrack "Cowbell"                  (X :| O :| X :| O :| X :| O :| X :| O)
```

Figure 1: Combining tracks

```
play :: Float -> Track -> IO ()
loop :: Float -> Track  -> IO ()
applyToMusic :: (Track -> Track) -> IO ()
```

Figure 2: HMusic primitives for live coding

but will loop the track in the BPM provided. If a loop is already being played, it will be substituted by the new one. The `applyToMusic` function can be used to modify the current pattern being played. It takes as argument a function from `Track` to `Track` and applies it to the pattern being looped.

These functions can be called in the Haskell interpreter (GHCi [12]) to live code music. Here is a simple example of a live code session. We start by looping a simple multi-track that contains only snare and kick:

```
*HMusic> loop 120 (kickTrack :|| snareTrack)
```

This call will start looping at 120 BPM a parallel composition of the `kickTrack` and `snareTrack` defined priviously in Section 2.2. Next, we can add to the loop being played another track with a hi-hat:

```
*HMusic> applyToMusic
(:|| MakeTrack "ClosedHiHat" (X:|X:|X:|X))
```

In this example, we are using partial application to transform the parallel composition operator (that has type `Track -> Track -> Track`) into a function that takes only one argument, i.e., `Track -> Track`.

Next, we can add a guitar sample in the beginning of the loop:

```
*HMusic> applyToMusic
(:|| MakeTrack "guitarSample" X)
```

The `map` function is omnipresent in functional programming languages and it is used to apply a function to all elements of a list. We can easily define a similar function for tracks:

```
mapTrack :: (Track -> Track) -> Track
            -> Track
mapTrack f (t1 :|| t2) =
     mapTrack f t1 :|| mapTrack f t2
mapTrack f t = f t
```

The `mapTrack` function can be used to modify tracks while they are being played. For example, we could write a function to substitute instruments of simple tracks:

```
subsInstr i1 i2 t@(MakeTrack i p)
  | i == i1 = MakeTrack i2 p
  | otherwise = t
subsInstr i1 i2 t = t

*HMusic> applyToMusic (mapTrack
(subsInstr "ClosedHiHat" "OpenHiHat"))
```

4.2 Effects in Live Coding

With HMusic, it is possible to dynamically add and remove effects from tracks being played. For example, it is possible to add effects to the multi-track being played:

```
*HMusic> applyToMusic
(\track -> Master [Amp 1.0] track)
```

Here we used a lambda abstraction to create a function that takes a track as an argument (the current track being played), and creates a master track around it.

As another example, we can write a function that substitutes a set of effects in a track by another set of effects:

```
subsEffects :: [Effect] -> [Effect]
               -> Track -> Track
subsEffects e1 e2 t@(MakeTrackE i e f)
  | e1 == e = MakeTrackE i e2 f
  | otherwise = t
subsEffects e1 e2 t = t
```

and use it to modify the parameter of an effect:

```
*HMusic> applyToMusic
(subsEffects [Reverb 1.0] [Reverb 0.3])
```

or to substitute a set of effects by another:

```
*HMusic> applyToMusic
(subsEffects [Reverb 1.0, Amp 1.0]
             [Attack 0.75, Release 0.75])
```

It is also possible to add effects to simple tracks by turning them into an effect track:

```
changeTrack :: Instrument -> [Effect]
               -> Track -> Track
changeTrack i e t@(MakeTrack it p)
  | i == it = MakeTrackE i e p
  | otherwise = t
changeTrack i e t = t
```

and use it to add an effect to a snare track:

```
*HMusic> applyToMusic
(changeTrack "snare" [Reverb 1.0])
```

We also provide a named master track:

```
data MTrack = (...)
  | MasterN  String [Effect] MTrack
```

Naming master tracks helps programmers to easily modify Master tracks (e.g., modify the effects being used, modify internal tracks by adding effects) while live coding. For example, we can define a named track for drums using the priviously definded `drums` track:

```
drumsN :: MTrack
drumsN = MasterN "drums" [Amp 0.2] drums
```

and while it is being played, we can modify its effects with the `changeMaster` function:

```
changeMaster :: String -> [Effect]
                -> Track -> Track
changeMaster name e t@(Master n em tm)
  | name == n = Master n e tm
  | otherwise = t
changeMaster i e t = t


*HMusic> applyToMusic
(changeMaster "drums" [Sustain 1.0])
```

```
trackcomp =
   MakeTrack "kick"                                                      X
   :|| MakeTrackE "snare" [Reverb 0.5]                                  (O :| O :| X)
   :|| MakeTrackE "hihat" [Attack 0.1, Sustain 0.3, Release 0.1] (X :| X :| X :| X)
   :|| MakeTrack "guitar"                                                X

live_loop :hmusic do
    sample :kick
    sample :hihat, attack: 0.1, sustain: 0.3, release: 0.1
    sample :guitar
    sleep 0.6
    sample :hihat, attack: 0.1, sustain: 0.3, release: 0.1
    sleep 0.6
    sample :snare, reverb: 0.5
    sample :hihat, attack: 0.1, sustain: 0.3, release: 0.1
    sleep 0.6
    sample :hihat, attack: 0.1, sustain: 0.3, release: 0.1
    sleep 0.6
end
```

Figure 3: A multi-track and its code representation in Sonic Pi

5 Compiling HMusic into Sonic Pi

The current implementation of HMusic compiles patterns and tracks into Sonic Pi [1] code. Sonic Pi is an educational programming language created with the objective of teaching programming to kids through the creation of music. It is an open source tool originally developed for the Raspberry Pi processor but it is also available for different platforms such as Windows, Linux and macOS. Although the tool has been initially designed for pedagogical purposes, it is currently being used by a variety of musicians for live coding performances.

To compile the abstractions provided by HMusic into Sonic Pi code, we use a small set of primitives provided by the language, such as loops, rests, playing sound samples and effects:

- `live_loop`: loops the sound generated by a set of Sonic Pi instructions
- `sleep n`: makes the current thread wait for n seconds (or a fraction) before playing the sound generated by the next instructions
- `sample :audofile`: plays the `audiofile`
- Effects: Sonic Pi provides a set of effects that can be applied to sound samples, e.g., Amp, Reverb, Sustain, Attack, Release, etc.

We use the BPM parameter of functions `loop` and `play` to calculate the time for each beat, e.g., for 100 BPM, 100 beats will be played in 60 seconds. Figure 3 shows a multi-track and the generated code in Sonic Pi for looping the track at 100 BPM.

In order to interact with the Sonic Pi server, we used the Sonic Pi Tool [13], which is a command line utility that allows to send messages to the Sonic Pi server without using its GUI interface. It provides commands to start the Sonic Pi server, stop it, and also allows to send code to be processed in real time. The Haskell's `System.Cmd` interface [14], which is a simple library to call external commands in Haskell, was used to implement the functions `play`, `loop` and `applyToMusic`. These functions transform HMusic tracks into Sonic Pi code, and use the `System.Cmd` library to access the sonic server tool and execute the generated music. When looping, the code for the current track being played is held in a global `IORef` [15], which is basically a pointer to the track being played. The `loop` function substitutes the track being played, and `applyToMusic` will modify it, compile it again, and send it to the Sonic Pi server through the Sonic Pi tool.

6 Related Works

There has been a lot of work on designing programming languages for computer music and live coding. Most of these languages, e.g., CSound [3], Max [4, 5], Pure Data [6], Supercollider [7], Chuck [8], FAUST [9] etc, are based on the idea of dataflow programming, where signal generators and processors can be connected either visually or through code, providing the abstraction of streams of data/-sound that can be combined and processed. Some languages for music programming e.g., Gibber [16] and IXI Lang [17], and music notation languages e.g., LilyPond [18] and abc notation [19], also provide ways of describing patterns and/or tracks, but do not focus on their composition/combination. In languages like Gibber and IXI Lang, patterns may contain different sounds, while in the abstraction provided by HMusic, patterns are associated with a single sound, just like in sequencers. Patterns in live coding languages can usually be routed to effects, while in HMusic effects are associated to individual tracks or multi-tracks.

There are many DSLs for computer music based on functional languages, e.g. [20, 21, 22, 23, 24]. These languages usually provide means for playing notes and composing the sounds generated in sequence and in parallel. In these languages the programmer can write a sequence of notes and rests, and these sequences can also be combined in parallel and applied to effects. In HMusic, instead of having different sounds in the same track, each track indicates when a single sound is played, i.e., It is the repetition pattern of a single sound, similar to what happens in grids of a drum machine and sequencers. Although the symbols used in HMusic have semantic meaning, visually programs look like an ASCII version of the grids for writing drum beats available in modern sequencers. We believe that this approach makes it easier for someone that is used with sequencer tools to write simple tracks in HMusic with little knowledge of functional programming. Furthermore, as patterns are not associated with sounds, patterns can be reused with different instruments when needed. HMusic is an extension of a language called HDrum [25]. HDrum is a language for drum beat programming, and is compiled into midi files. No loading of samples or live coding is supported.

The formal semantics of a language with support for live coding is the subject of Aaron et. al. work [26]. The authors discuss some problems with the semantics of Sonic-Pi `sleep` function and propose a formalization to fix the problem while being compatible with Sonic-Pi previous versions. The work introduces the notion of *time-safety* and shows that Sonic-Pi's new semantics is time-safe. Time safety is an important notion when programs consists of multiple threads that need to cooperate to produce a music. Since HMusic semantics is compiled to Sonic Pi, it enjoys the time safety property. We let the formalization of HMusic compilation process and its extension to support multi-thread programs, like Sonic-Pi, to future work.

7 Concluding Remarks

This paper described how to extend HMusic with abstractions for effects. Two new types of tracks were added to the language, a track that allows the application of effects on patterns and a track that allows effects on multi-tracks. The implications of track composition/combination in the presence of effects were also discussed. HMusic provides a small set of primitives for playing music and also for live coding, and we demonstrated that the new abstractions for effects can also be manipuled on the fly in a live code session.

Development of music and live coding in HMusic would be much easier with a special editor that could, either visually or with options in a menu, generate automatically empty tracks of a desired size, with the programmer being responsible for filling the hits. One simple way of obtaining such a feature is using Emacs macros [27]. The system implemented to support HMusic could be easily extended for collaborative live coding, where different programmers interact with the music at the same time. HMusic tracks can be converted into strings of text using Haskell's `Read` and `Show` type classes [28], hence a simple interface for collaborative live coding can be obtained with a socket server that receives code, which is processed locally in the clients, and sends to be run on a central Sonic PI server. Elm [29], is a functional programming language with syntax and many features similar to Haskell. It is compiled into JavaScript, and used to create web browser-based graphical user interfaces. We believe that HMusic could easily be ported to Elm which would allow web-based music performances.

Acknowledgments

This work was supported by CAPES/Brasil (Programa Nacional de Cooperação Acadêmica da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

References

[1] Samuel Aaron, Alan F. Blackwell, and Pamela Burnard. The development of Sonic Pi and its use in educational partnerships: Co-creating pedagogies for learning computer programming. *Journal of Music, Technology and Education*, 9:75–94, 05 2016.

[2] The HMusic DSL. https://github.com/hmusiclanguage/hmusic, January 2019.

[3] CSound. http://csound.com, January 2019.

[4] Emmanuel Favreau, Michel Fingerhut, Olivier Koechlin, Patrick Potacsek, Miller Puckette, and Robert Rowe. Software developments for the 4x real-time system. In *International Computer Music Conference*, 1986.

[5] David Zicarelli. How I Learned to Love a Program That Does Nothing. *Computer Music Journal*, 26(4):44–51, 2002.

[6] Miller Puckette. Pure Data: another integrated computer music environment. In *in Proceedings, International Computer Music Conference*, pages 37–41, 1996.

[7] James McCartney. Supercollider, a new real time synthesis language. In *International Computer Music Conference*, 1996.

[8] Ge Wang and Perry Cook. Chuck: A programming language for on-the-fly, real-time audio synthesis and multimedia. pages 812–815, 01 2004.

[9] Yann Orlarey, Dominique Fober, and Stephane Letz. FAUST: an Efficient Functional Approach to DSP Programming. In *New Computational Paradigms for Computer Music*, 2009.

[10] Christopher Alex McLean. *Artist-Programmers and Programing Languages for the Arts*. PhD thesis, University of London, 2011.

[11] Andre Rauber Du Bois and Rodrigo Geraldo Ribeiro. Hmusic: A domain specific language for music programming and live coding. In Marcelo Queiroz and Anna Xambó Sedó, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 381–386, Porto Alegre, Brazil, June 2019. UFRGS.

[12] Glasgow Haskell Compiler. https://www.haskell.org/ghc/, January 2019.

[13] Sonic Pi Tool. https://github.com/lpil/sonic-pi-tool, January 2019.

[14] System.Cmd. http://hackage.haskell.org/package/process-1.6.5.0/docs/System-Cmd.html, January 2019.

[15] Simon Peyton Jones. Tackling the Awkward Squad: monadic input/output, concurrency, exceptions, and foreign-language calls in Haskell. In *Engineering theories of software construction*, pages 47–96. Press, 2002.

[16] Charles Roberts, Matthew K. Wright, and JoAnn Kuchera-Morin. Music Programming in Gibber. In *ICMC*, 2015.

[17] Thor Magnusson. The IXI Lang: A SuperCollider Parasite for Live Coding. In *International Computer Music Conference*, 2011.

[18] LilyPond. http://lilypond.org/, January 2019.

[19] ABC Notation. http://abcnotation.com/, January 2019.

[20] Alex McLean. Making Programming Languages to Dance to: Live Coding with Tidal. In *FARM 2014*. ACM, 2014.

[21] Paul Hudak, Tom Makucevich, Syam Gadde, and Bo Whong. Haskore music notation: An algebra of music. *J. of Functional Programming*, 6(3), May 1996.

[22] H. Thielemann. Audio Processing Using Haskell. In *DAFx04*, 2004.

[23] Paul Hudak and David Janin. Tiled polymorphic temporal media. In *FARM 2014*. ACM, 2014.

[24] Paul Hudak. An algebraic theory of polymorphic temporal media. In *PADL*, 2004.

[25] André Rauber Du Bois and Rodrigo Geraldo Ribeiro. A domain specific language for drum beat programming. In *Proceedings of the Brazilian Symposium on Computer Music*, 2017.

[26] Samuel Aaron, Dominic Orchard, and Alan F. Blackwell. Temporal semantics for a live coding language. In *Proceedings of the 2Nd ACM SIGPLAN International Workshop on Functional Art, Music, Modeling & Design*, FARM '14, pages 37–47, New York, NY, USA, 2014. ACM.

[27] The Emacs editor. https://www.gnu.org/software/emacs/, January 2019.

[28] Simon Thompson. *The Craft of Functional Programming*. Addison-Wesley, 2011.

[29] The Elm Programming Language. https://elm-lang.org/, January 2019.

Prototyping web instruments with Mosaicode

André Lucas Nascimento Gomes , Frederico Ribeiro Resende , Luan Luiz Gonçalves , Flávio Luiz Schiavoni

[1] Arts Lab in Interfaces, Computers, and Everything Else - ALICE
Federal University of São João del-Rei – UFSJ
São João del-Rei - MG - Brazil

fredribeiro97@gmail.com,andgomes95@gmail.com

luanlg.cco@gmail.com,fls@ufsj.edu.br

Abstract. *Many HTML 5 features enable you to build audio applications for web browsers, simplifying the distribution of these applications, and turning any computer, mobile, and portable device into a digital musical instrument. Developing such applications is not an easy task for lay-programmers or non-programmers and may require some effort by musicians and artists to encode audio applications based on HTML5 technologies and APIs. In order to simplify this task, this paper presents the Mosaicode, a Visual programming environment that enables the development of Digital Musical Instruments using the visual programming paradigm. Applications can be developed in the Mosaicode from diagrams – blocks, which encapsulate basic programming functions, and connections, to exchange information among the blocks. The Mosaicode, by having the functionality of generating, compiling and executing codes, can be used to quickly prototype musical instruments, and make it easy to use for beginners looking for learn programming and expert developers who need to optimize the construction of musical applications.*

1 Introduction

Recently, with the emergence of HTML 5 and the Web Audio API, the web browser became a feasible environment to host a DMI (Digital Musical Instrument). Beyond the acclaimed Web Audio API, HTML 5 brought several others technologies useful to DMIs development like the Web MIDI API, WebRTC, Gamepad API, Websockets, SVG Canvas, WebGL, new tags and elements. Together, all these technologies can be a really powerful framework to develop new DMIs. In order to merge the technologies behind HTML 5, unleashing the development of DMIs for the web, this paper presents the Mosaicode, a Visual Programming Environment that can help novices and experts programmers in implementing and prototyping applications on the field of Digital Arts – better detailed in Section 3.

Several programming tools and languages already support the development of DMIs, including a technological apparatus to help programmers, non-programmers and lay-programmers to perform this task. A non-exhaustive list of tools and languages to develop new DMIs is presented in Section 2.

Different from other related tools, the Mosaicode is an application to generate code and complete applications using the Visual Programming paradigm to the Digital Arts domain.

This tool can be complemented by extensions, allowing one to add resources that define a Visual Programming Language (VPL) for developing applications for new domains. Among all extensions of Mosaicode, there is an extension to develop Web Art applications that supports the creation of DMIs. This extension has several blocks of code to access physical and logical inputs, audio sources and effects, MIDI devices, HTML elements and more. The development of DMIs using Mosaicode is presented in Section 4, including examples of DMIs developed in the environment.

The Mosaicode has already being used as a support tool to the course “Introduction to Computer Music” on the Computer Science Department at the Federal University of São João del-Rei – UFSJ. The students developed DMIs in an easy way, just by dragging and connecting blocks, focusing only on the concept of DMIs development. It was really interesting to use the Web Art extension to this purpose. Mosaicode also was used to create instruments to a live performance with the audience participation developed in our research group. This experiences and discussion about the use of Mosaicode is presented in Section 5. At the end, Section 6 presents some final considerations.

2 Related works

Currently there are several musical programming languages and tools, with different aspects and paradigms, to help the creation of new DMIs.

Pure Data[1] (a.k.a. Pd) is visual programming environment developed to create real time sound and music projects. This open source tool was developed by Miller Puckette in the 90’s [1]. Although its main focus on audio manipulation, Pd enables to work with data from different sources that can be treated in an interconnected way. As a consequence, it facilitates the coupling of audio, video and MIDI applications, among others, that the tool supports.

Max/MSP [2] is, like Pd, a visual programming environment for image, sound and video processing [2], commonly used by artists, performers and composers to create their applications. Different from Pd, Max is a proprietary software.

[1] Available on `http://puredata.info`.
[2] Project Website: `https://cycling74.com/products/max`.

EyesWeb[3] is another visual development environment. This is specifically an open-source programming tool for image, sound and video, with an emphasis on gesture analysis and body movement [3]. A major advantage of it is the number of input devices supported, such as cameras for image capture, video game accessories like Kinect and Wii controls and multi-channel audio.

ISADORA[4] is an interactive tool for media (sound and video) that is widely dedicated to artists and performers looking for the creation of pieces and performances by iterating image and sound, allowing real-time processing and customization [4]. This environment has advantages such as appearance and friendly interface, along with the practicality of visual programming; in addition, it is proprietary software.

Processing[5] is an open-source textual programming language for the Digital Art domain developed by the MIT Media Lab [5]. This language is used in a didactic way for teaching programming to facilitate, captivate and bring up new students to the area of software development.

CodeCircle[6] is a software that provides a real-time, collaborative and social coding by means a web prototype environment for musically expressive instrument development. The user can implement applications using a specific code, including HTML, CSS and JavaScript by the web interface that consists of the code editor and the resulting web. For generate sound in the browser and to implement interactive machine learning, this software uses the MaxiLib and RapidLib libraries, respectively. Expressive interactions are designed using "programming by demonstration" [6].

JythonMusic[7] supports computer-assisted composition by a free and open source environment based on Python for interactive musical experiences and application development. The use of the Jython programming language enable to work with Processing, Max/MSP, Pure-Data and other environments/languages, and also giving access to Java API and Java based libraries. It's allowed to the user to interact with external devices such as MIDI, manipulate images and also create graphical interfaces [7].

FAUST[8] is a functional programming language for sound synthesis and audio processing. A code developed in FAUST can be translated to a wide range of non-domain specific languages such as C++, C, Java, JavaScript, LLVM bit code, and WebAssembly[8]. There is a web platform called "Faust Playground" designed to enable children to learn basic audio programming, providing graphics resources to create DMIs.

Certainly, all these related tools have some difference with the tool presented in this paper and all of them are more mature and solid tools to create audio and music applications. Some of them are also visual, some are code generators. The most important to say about all these tools is that they all inspired our team to develop our programming environment and there are much more to learn about them to evolve our tool.

[3]Project Website: `http://www.infomus.org/`.
[4]Project Website: `https://troikatronix.com/`.
[5]Project Website: `https://processing.org/`.
[6]Project Website: `https://codecircle.gold.ac.uk/`.
[7]Project Website: `http://jythonmusic.org`
[8]Available on `https://faust.grame.fr/`

3 Mosaicode

Mosaicode [9] is a visual programming environment focused on the development of applications in the specific domains of Digital Arts. This programming environment, depicted in Figure 1, is a Desktop application developed in Python that provides elements to create applications for Digital Art domain involving Computer Science topics like Artificial Intelligence, Virtual Reality, Computer Graphics, Computer Vision and Computer Music.

This environment is an open-source code generation tool that aims to unleash development of applications and fast prototyping. Thus, it can be quick and simple for those who do not have the programming skill to create an application and for expert programmer to fast prototype, to fast change code – can quickly test new code settings – and also to change/optimize the application directly by the generated code. Based on this, the Mosaicode can be used to a fast start a project generating the code of a working prototype. The simplicity of the environment is based on the Visual Programming paradigm, used by several other related tools and very common on Digital Arts domain. An application in the Mosaicode is created dragging, dropping and connecting blocks.

A **Block** is the most basic unit in the environment and it is responsible to perform minimal, atomic and specific activity within a domain. A Block can have different behaviours and it can be set up by the Block's properties. Besides, a **Property** can be set up statically or dynamically. A static property is a parameter that influences the Block's execution and that can be modified in programming time, however, it is constant during application running. Figure 1.C presents the side bar to set up a Block's static properties in the environment.

A dynamic property uses a input **Port** of the Block to change the Block's settings and it is done at run time. A Block can also have output ports, to send values to other blocks. Thus, using these typed ports and connections, is possible to exchange values between blocks. A set of these blocks connected together is called **Diagram**.

A Diagram, like presented in Figure 1.D, is used to generate an application code based on a **Code Template**, merging code snippets and creating the final application.

By default, when there is a data stream on the input ports, these values override the static properties associated with each input port. Thus, when connecting a block, which outputs information from a sensor, to the input of a block that performs an arithmetic operation, for instance, the arithmetic block will no longer use the value assigned in the static property, it will use the sensor values to perform the operation.

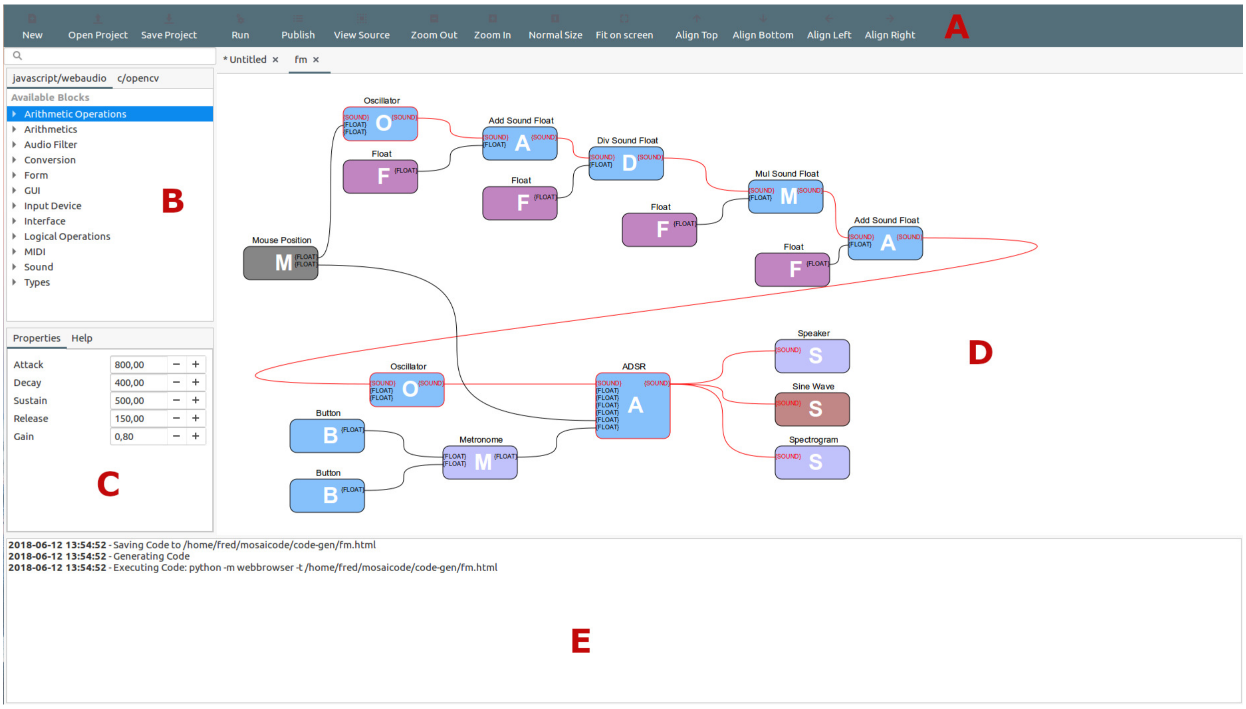


Figure 1: It shows the graphical interface of the Mosaicode.

When implementing a block diagram in Mosaicode, users do not have to worry about remind programming languages commands and syntax, they just need to be aware of the specific domain to know which blocks must be used and how to connect them to generate the desired application. Once a user acquires knowledge about the domain, it becomes easier to open the source code to study it and to learn how the application is implemented using a particular programming language.

3.1 Extending the environment

A set of blocks, ports, and code templates composes a Mosaicode **extension**. An extension can generate source code/applications for a particular programming language and a specific domain.

Currently, Mosaicode has some extensions to generate application code in C/C++ language for subjects like Sound Design, Digital Image Processing, Image Synthesis, GUI, Joystick Control and Computer Vision. Each one of these extension use an external library to support the application development, such as OpenCV for Computer Vision and Digital Image Processing [10], GTK for GUI creation, openGL to image synthesis and the PortAudio for Sound Design [11].

This environment also has an extension for Web Art development that generates HTML 5 + CSS + JavaScript application code. This extension is explained in the next section, discussing about how it can be used to create DMIs.

4 DMI development with Mosaicode

To present the DMI development with Mosaicode, we are adopting a common vision that a DMI can be splitted in three parts: the input, that captures gestures of the musician and involves the physical and virtual interactions of the user; the output, responsible for synthesizing the sound of the instrument and give other feedback to the user like visual and haptic; and the mapping, a strategy to interconnect the input with the output. Figure 2 presents this schematic, as will used in next Sections.

4.1 User Input

User input is intended to receive data streams referring to user interactions with the application. User inputs can be performed by physical devices, like sensors, or graphical interfaces components, like buttons, input text boxes and sliders. A computer mouse and keyboard are common devices that can be used to interact with applications and that can use GUI elements to intermediate this interaction. Beyond GUI elements, the JavaScript programming language easily provides resources to capture mouse events like click and movements and keyboard events like key press and release. Other input devices connected to the computer can also be accessed using HTML 5, specially when using smartphones and tablets. Using HTML5, the GPS position can be reached using the Geolocation, Joystick and game devices can be accessed with the GamePad API and MIDI devices can be accessed by the Web MIDI API (physical or virtual). Other sensors and physical devices like touch screen, gyroscope and accelerometers can also be accessed and used as user input using the JavaScript language. Javascript also allows to access the camera and microphone using the WebRTC API in real-time [12, 13].

GUIs designed to provide communication between users and applications (user input) are composed of HTML 5 elements. These elements can be: text fields, buttons, slider controls, number field, radio, check and others.

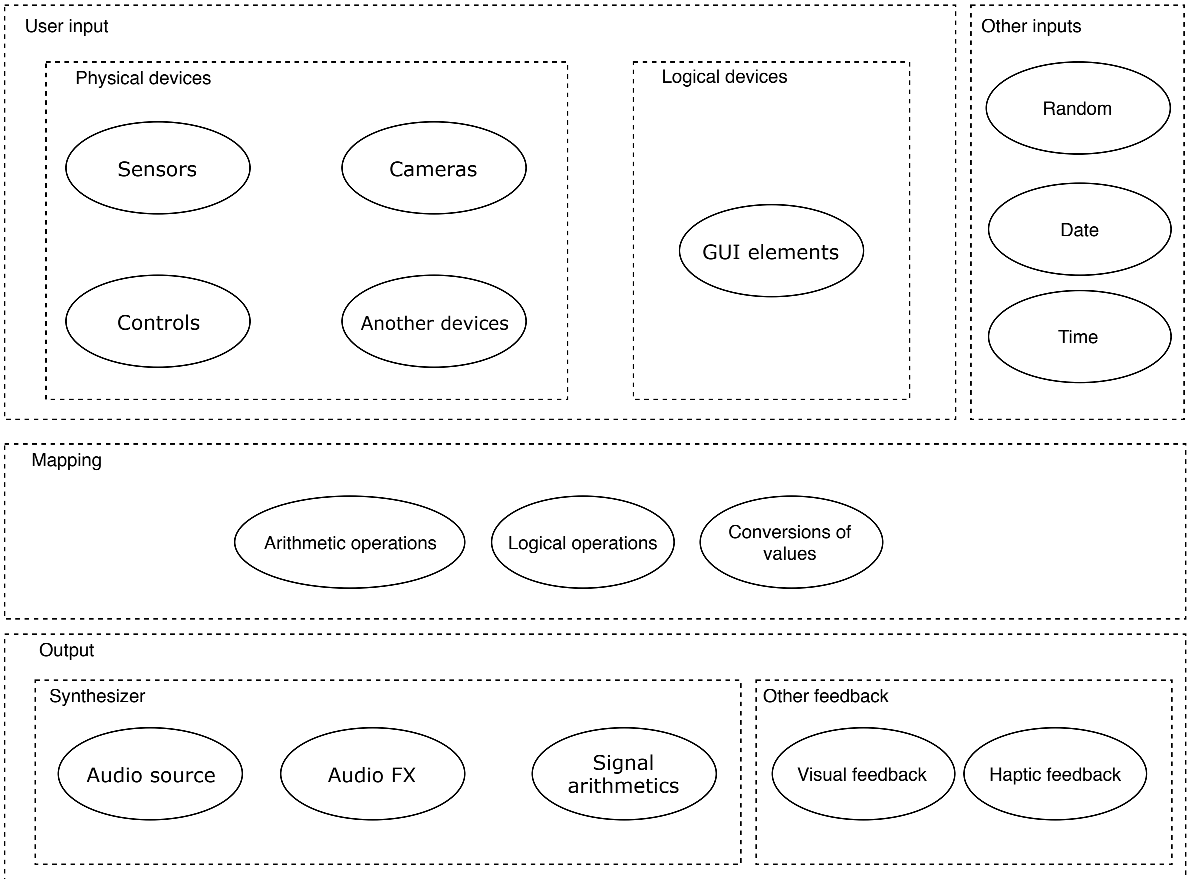


Figure 2: Basic structure of DMI.

GUIs are interesting to control DMIs for their accessibility, not requiring the purchase of other devices to play them.

Table 1: List of Mosaicode's input blocks to generate Web Audio applications in HTML 5/JavaScript language.

Categories	Blocks
Form	Button, Check, Number, Range, Select, Text.
Input Device	Date, Device Orientation, Hour, Keyboard, Microphone, Mouse Click, Mouse Position, Orientation Change, Up Keyboard.
Interface	Increment, Metronome, Random.
MIDI	MIDI in
Sound	Microphone

In our Web Art extension, we have joined the APIs resources and made them into Mosaicode blocks with the same functionality, enabling many ways for the application to receive data by input user. This extension blocks were organized by categories to make it easier to find the desired features for developing applications, as presented in Table 1

4.2 Other Inputs

There are other sensors and values, available in devices, that can be used as input in DMI development, but that are not controlled by user actions. These devices are the computer clock, providing date and time, and random values, for instance. This input can be used to add some stochastic parameters to DMIs, schedule events, delay events and create sequencers. All these functionalities are native in JavaScript programming and also became blocks to our environment.

We also implemented Numbers and other constants Blocks to use it as value input in diagrams. It is very useful to set up fixed values and normally used to set up initial values to run the application.

4.3 Synthesizer

The synthesizer is the voice of the DMI. Normally, a synthesizer can implement or be inspired by one or more classic algorithms of audio synthesis like AM, FM, PM, additive, subtractive, physical modelling, and others.

The Web Audio API provides several elements to create a synthesizer like Oscillators, Gain, Filters, Audio Spacialization, and also audio FX like delay, reverb, chorus, phaser, flanger, distortion, filters and others. Thus, to implement a synthesizer with this API is really simple and depends on computer music knowledge of how to create synths. All these elements were implemented as Blocks in the Mosaicode environment.

Some audio operations that are not available in Web Audio API but are important to create synthesizers were implemented using the Web Audio ScriptProcessorNode. Arithmetic operations to audio signals, White

Noise and ADSR envelope are examples of resources implemented to Mosaicode as Blocks using this feature of the Web Audio API. A not complete list of the Blocks to create a sound synthesizer is presented in Table 2.

Table 2: List of Mosaicode's blocks to generate web audio applications in HTML 5/JavaScript language.

Categories	Blocks
Audio Filter	Allpass, Bandpass, Highpass, Highshelf, Lowpass, Lowshelf, Notch, Peaking.
Sound	Add Sound, Add Sound Float, ADSR, Channel Merge, Delay, Distortion, Divide Sound, Gain, HRTF, Multiply Sound, Multiply Sound Float, Oscillator, Playback, Speaker, Subtract Sound, Subtract Sound Float, White Noise.

4.4 Other Feedback

A DMI can also use another outputs to help the user to understand its behavior. Our first explored output is a visual feedback, the resource to print a value on the web page; maybe it is the most used Block to debug code. To give visual feedback to user and to create a nice design it is also possible to change the background color of the page and other page elements, like the page title.

Some more interesting visual feedback were created using the HTML 5 Canvas element to draw representations of audio signals. There are Blocks to show a frequency bar chart, waveform and audio spectrum. These blocks also use the WebGL API, which supports rendering of 2D/3D graphics in real-time HTML Canvas elements, enabling analysis of audio signals at runtime [14].

Another possible output is to create and MIDI virtual device to output values from the web application to a local synthesizer, logical or physical, using the WebMIDI API.

We can use another two interesting resources to create feedback: the webcam flash light, accessible by WebRTC and the vibracall, using the Vibration API. Using the flash provides visual feedback such as graphics, background color and page title, and the vibrating alert provides tactile feedback.

All these possibilities are implemented in our extension and available in Blocks, like present in Table 3.

Table 3: List of Mosaicode's blocks to give feedback to user.

Categories	Blocks
HTML	Title, background color, print
Canvas	Frequency Bar, Print, Sine Wave, Spectogram
MIDI	MIDI Output
Mobile Output	Flash, Vibracall

4.5 Mapping

Although mapping does not uses to be a resource that can became a block , during the implementation of syntheses and audio effects, sometimes it became necessary to map values by adjusting them to vary in a certain range. For example, the position of a mouse on the screen, ranging from 0 to 1024, can be mapped to a gain, ranging from 0 to 1. Thus, it became necessary to have some blocks of mathematical and other operations to be used to convert values from the input to the output.

We created a set of Blocks to perform operations between float numbers, logical operations and to decrement and increment values. There are also a set of conversion blocks provided in Mosaicode to strip MIDI values and to convert MIDI notes to frequency, to convert from float to RGB, RGB to Float, Float to Boolean and Float to Char.

These elements are delegated by the mapping unit as parameters of DMI synthesis. In this way, joystick buttons, hand mapping with cameras and accelerometers and cell phone gyroscopes can be used as inputs to the synthesizing, controlling elements such as note duration and frequency, filter frequency, general gain and noise gain. Table 4 presents some Blocks that can be used to mapping values.

Table 4: List of Mosaicode's mapping blocks to generate web audio applications in HTML 5/JavaScript language.

Categories	Blocks
Logic	Equals To, Greater Than, Greater Than Or Equals, Less Than, Less Than Or Equals, Not Equals To.
Arithmetics	Add Float, Divide Float, Max Float, Modulos, Multiply Float, Subtract Float.
Conversion	Bool To Float, Char To Float, RGB.
MIDI	MIDI To Frequency, Strip MIDI.

4.6 Examples

An example of a FM synthesizer developed in Mosaicode is presented in Figure 3. In this example, the carrier oscillator (sine wave format) has as initial frequency value 440 Hz and the frequeency of the modulator oscillator in 4000 Hz.

The values generated by the oscillators range from -1 to 1. To change this value to a range of 220 to 2220, we added the value 1 to the oscillator output, making it vary in the interval from 0 to 2. Then we multiplied it by 1000 doing it vary from 0 to 2000 and finally we added the value 220 to vary the value from 220 to 2220. Thus, we have a periodic frequency change in the range of 220 Hz to 2220 Hz occurring 4000 times per second, which determines the carrier oscillator frequency over that time. The carrier oscillator output was directed to the Sine Wave block, to be drawn in the waveform as shown on Figure 4, and also directed to the Speaker block.

Another example is presented in Figure 5a. This DMI uses a ADSR envelope receiving a white noise as input and being triggered by a button. Finally, the envelope output is connected to a Speaker and a frequency bar.

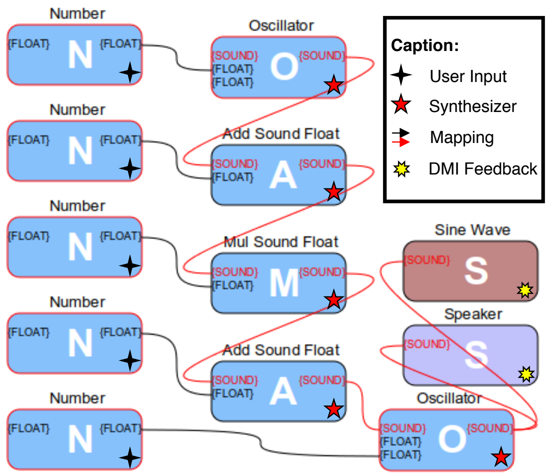


Figure 3: FM synthesis using the Mosaicode-javascript-webaudio extension.

Figure 4: Visualization of waveform generated by the FM Synthesis on the example of Figure 3.

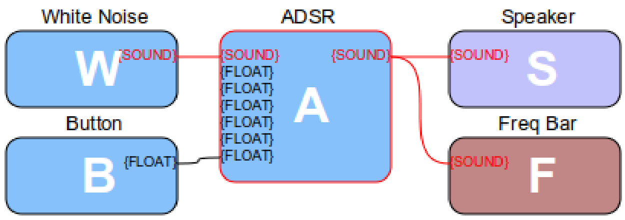
So, when running the application, we have a button on the web interface (5b), the frequency bar chart, which varies when playing the instrument, and we can hear the instrument sound.

These examples use numeric fields to control synthesizer parameters and a button to dispatch the ADSR envelope. It is possible to replace this fields with any other user input block just changing it in the diagram or dispatch the white noise's envelope with a metronome, for instance.

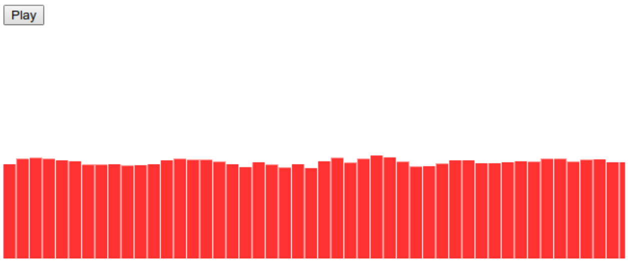
5 Discussion

We used Mosaicode as a support tool to implement and prototype DMIs in a Computer Science course called "Introduction to Computer Music" [15] that had the main audience undergrad students on Computer Science field. In these classes we noticed how important is to have a development environment that enables rapid prototyping and creation, also to initiate instrument designs. Often, a synthesizer is born from experimentation with signal arithmetic, testing and experimenting with settings and parameters. The same can occur with the choice of interfaces or mappings.

At least during prototyping or in the classroom, constructing a DMI resulted in ephemeral codes, an experimentation that can be reused, but that also can be simply discarded due to this ephemerality. Throwing work away may not be a problem when we know that doing it again can be simple and even fun.



(a)



(b)

Figure 5: Diagram with White Noise controlled by an ADSR envelope, with outputs on a Speaker and a Freq Bar, shown in (b).

We also used it in the context of DMI development, with audience participation in a multimedia performance called Chaos das 5. In this performance, the audience could access web instruments and take part of the soundscape of the performance. The development of these instruments by our research group was made improvising and playing sounds, sometimes totally free, based on experimentation and trials. We used pair programming including students with different levels of knowledge in Computer Music and synthesis algorithms. However, after a few meetings, all students could already create sounds and develop DMIs, even without a formal course in Music Computing.

6 Conclusion

This paper presented the Mosaicode, a visual programming environment for the domain of Digital Arts that here was explored to create DMIs using a Web Art extension. This extension, based on several APIs from the HTML5, can be used to create really interesting and multi-platform DMIs that would be hard to code using JavaScript directly. The development of this extension on Mosaicode can offer the power of HTML 5 by the means of a visual programming environment.

It can bring several advantages, such as rapid prototyping, as well as practicality, easy experimentation, trials and combination of blocks, generating completely different applications. It is also possible to use it as a support tool to teach lay programmers and artists to develop their applications without requiring learning a textual programming language. For this, Mosaicode offers to the user a wide range of possibilities and different combinations of Blocks.

For future work, we intend to maintain the developed extension and to review the set of blocks, adding new features whenever possible to make the extension more complete. In addition, an extension for MIDI controls is being developed, with various input types available to different media, using the C language. Initially, the MIDI category of audio synthesis extensions is scarce and limited, we have the intention of merging the extensions of the same programming language in order to rapidly expand the possibilities of development, allowing the user to integrate different domains in a simple way.

When creating DMIs it was important to think about communication interfaces that offer a good musical expression. For this, we implemented blocks that allow the use of external devices to control the synthesizers with a certain degree of complexity, trying to reach a more expressive DMI [16].

7 Acknowledgments

Authors would like to thanks to all Arts Lab in Interfaces, Computers, and Everything Else (ALICE) members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq, FAPEMIG, and also the PROAE/PROPE/UFSJ for the institutional support.

References

[1] Miller S Puckette et al. Pure data. In *ICMC*, 1997.

[2] Matthew Wright, Richard Dudas, Sami Khoury, Raymond Wang, and David Zicarelli. Supporting the sound description interchange format in the max/msp environment. In *ICMC*, 1999.

[3] Antonio Camurri, Shuji Hashimoto, Matteo Ricchetti, Andrea Ricci, Kenji Suzuki, Riccardo Trocca, and Gualtiero Volpe. Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal*, 24(1):57–69, 2000.

[4] TroikaTronix. Isadora. `https://troikatronix.com/`. Acessado em: 2018-07-29.

[5] Casey Reas and Ben Fry. *Processing: a programming handbook for visual designers and artists*. Mit Press, 2007.

[6] Michael Zbyszynski, Mick Grierson, Matthew Yee-King, et al. Rapid prototyping of new instruments with codecircle. In *Proceedings of the international conference on new interfaces for musical expression*. NIME, 2017.

[7] Bill Manaris, Blake Stevens, and Andrew R Brown. Jythonmusic: An environment for teaching algorithmic music composition, dynamic coding and musical performativity. *Journal of Music, Technology & Education*, 9(1):33–56, 2016.

[8] Yann Orlarey, Dominique Fober, and Stéphane Letz. Faust: an efficient functional approach to dsp programming. *New Computational Paradigms for Computer Music*, 290:14, 2009.

[9] Flávio Luiz Schiavoni, Luan Luiz Gonçalves, and José Mauro da Silva Sandy. Mosaicode and the visual programming of web application for music and multimedia. *Revista Música Hodie*, 18(1):132–146.

[10] Gary Bradski and Adrian Kaehler. Opencv. *Dr. Dobb's journal of software tools*, 3, 2000.

[11] Ross Bencina and Phil Burk. Portaudio-an open source cross platform audio api. In *ICMC*, 2001.

[12] Hongchan Choi and Jonathan Berger. Waax: Web audio api extension. In *NIME*, pages 499–502, 2013.

[13] David B Ramsay and Joseph A Paradiso. Grouploop: a collaborative, network-enabled audio feedback instrument. In *NIME*, pages 251–254, 2015.

[14] John Congote, Alvaro Segura, Luis Kabongo, Aitor Moreno, Jorge Posada, and Oscar Ruiz. Interactive visualization of volumetric data with webgl in real-time. In *Proceedings of the 16th International Conference on 3D Web Technology*, pages 137–146. ACM, 2011.

[15] Flávio Luiz Schiavoni Schiavoni, Thiago Thadeu Souto Cardoso, André Lucas Nascimento Gomes, Frederico Ribeiro Resende, and José Mauro da Silva Sandy. Utilização do ambiente mosaicode como ferramenta de apoio para o ensino de computação musical. In *Proceedings of 8th workshop on ubiquitous music (UbiMus)*, 2018.

[16] Christopher Dobrian and Daniel Koppelman. The'e'in nime: musical expression with new computer interfaces. In *Proceedings of the 2006 conference on New interfaces for musical expression*, pages 277–282. IRCAM—Centre Pompidou, 2006.

PSYCHO library for Pure Data

Alexandre Torres Porres [1]

[1] EL Locus Solus, independent center for research and computer music
Rua Doutor Eduardo de Souza Aranha 122 ap 181 045.43.902 São Paulo, SP

el.locus.solus@gmail.com

Abstract

This paper describes the PSYCHO library for the Pure Data programming language. This library provides novel functions for Pure Data and is a collection of compiled objects, abstractions and patches that include psychoacoustic models and conversions. Most notably, it provides models related to Sensory Dissonance, such as Sharpness, Roughness, Tonalness and Pitch Commonality.

This library is an evolution and revision of earlier research work developed during a masters and PhD program. The previous developments had not been made easily available as a single and well documented library. Moreover, the work went through a major overhaul, got rid of the dependance of Pd Extended (now an abandoned and unsupported software) and provides new features.

This paper describes the evolution of the early work into the PSYCHO library and presents its main objects, functions and contributions.

1. Early work

The work presented on this paper started with an older research on the implementation of psychoacoustic models for measuring Sensory Dissonance. The starting point was the implementation of a roughness model during a masters research [1], motivated by its adoption in measuring the dissonance of musical intervals taking into account the spectrum (as developed by Sethares [2]). This was first implemented as a Pure Data patch [3] — more precisely, as a Pd Extended patch, as it needed to run in Pd Extended and relied on some of its built in externals.

During a following PhD research, in 2009, a compiled external for both Pure Data and MAX was first developed instead of just a patch [4]. A compiled external provides much more CPU efficiency, which is highly needed for a live electronics computer music system such as Pure Data, where you can analyze sounds in realtime for a live performance.

This single external that included only one roughness estimation class evolved into a library as a single binary pack by the end of the PhD research, but for Pd only [5] (no Max externals have been provided ever since, and the old one is not available anymore). This library contained other classes because the model got more complex and included more features that could also be instantiated as separate objects — such as the conversion from linear amplitude to phons.

In a similar fashion, a library of compiled objects as a single binary pack was developed to include classes related to the Pitch Commonality model as described by Parncutt & Strasburger [6], including the Tonalness model — these had also been first implemented as Pd Extended patches.

This library, unlike the roughness library, had a modular structure, where the Pitch Commonality model needed to be constructed with a few objects from the library, but the same objects could also be used to derive a Tonalness model, for instance.

Besides these and other compiled objects, a few patches for Pd Extended were also developed during the PhD research (some of which relying on the externals developed by the author). This body of work has been discussed in previous papers, but as a toolbox [7] and not as the “PSYCHO library”. This early work was then available for Pd Extended and only binaries for macOS were provided.

It is important to note that the outcome of the PhD research was not only the development of computer music tools. It also included an important creative front as a big priority, which applied the provided tools in artistic works and discussed them. Moreover, the thesis provides an extensive and critical review of the psychoacoustic theory, including a psychoacoustic study to better investigate the modeling of psychoacoustic dissonance. Above all, a special attention was given to the review of Roughness modeling theory.

Therefore, the development and implementation of computer music tools during the masters and PhD was just one of the main pillars of the research. An original contribution was made with the implementation of an original roughness external, plus the first implementations of Pitch Commonality and other objects for Pure Data. Nonetheless, as usual in software development, further developments and enhancements were planned for the future.

2. Blowing the dust off and revision of the work

The author lacked technical skills to work independently in coding compiled objects, so a collaboration with others had been necessary for the first developments. Recently, the author has acquired self sufficiency and has worked in the development of other Pure Data libraries[1]. Hence, relying on collaborators is not an obstacle anymore.

1 See <https://github.com/porres/pd-else> and <https://github.com/porres/pd-cyclone>.

In the meantime, the project was left in the dust, and an issue imposed by the passage of time was the fact that Pd Extended got abandoned and unsupported (and every example patch and abstraction developed relied on it). This isn't that impeditive as one can now easily install libraries from Pd Extended in the main distribution of Pure Data (a.k.a Pd Vanilla). Nonetheless, the most elegant and easiest solution is to offer an independent Pd library and a body of work that doesn't rely on third party external libraries to run.

This basically required a complete rewrite from scratch to get rid of all externals dependencies and stick to Pd Vanilla (making use of its new features and objects). Besides the example patches, most of the models and compiled objects had also been implemented as Pd Extended patches. As a result, these are now also implemented as pure Pd vanilla patches.

As a more technically fluent programmer, all of the code got also rewritten and revised. In this process, the overall design of the objects went through major surgery as well. As the result, all objects are now available as a separate/single compiled binary, instead of a binary pack that includes many objects as before. This design choice is considered best as it's just more common for Pd libraries to come as separate binaries these days.

The code of each object is also separate now. Hence, the modular structure of the Pitch Commonality model was abandoned and now there's just a single object to measure Tonalness, for example. This major overhaul makes the early work completely obsolete, and is now completely unavailable and lost to oblivion. Besides that, there's also the inclusion of new objects and developments.

3. The PSYCHO Library for Pd

This section of the paper describes the main features of the PSYCHO library. At the time of this writing, there's an initial 1.0 release that inaugurates this new incarnation phase of the research. It has been tested and runs on Pd Vanilla 0.50 or later. It is available on GitHub[2] and you can also install it via Pd's external manager (a.k.a. 'deken', under Help => Find Externals), just search for "psycho".

3.1 Roughness

Roughness is the main dimension of Sensory Dissonance since Helmholtz [8]. The main reference for the provided Roughness model is the work by Clarence Barlow [9], but a revision was proposed to include Vassilakis' amplitude fluctuation degree [10]. The object also offers the complete model by Vassilakis and its main reference (the model of Sethares [2]). You can also tweak the parameters independently to investigate the difference between these parameters and models. The [roughness] object takes as an input a list of frequencies and amplitudes to estimate the roughness.

One of its applications is to draw roughness curves to derive musical scales according to a spectrum. The [roughcurve] object is an abstraction that uses the [roughness] external to draw dissonance curves (with roughness estimation in the vertical axis and interval in cents on the horizontal axis). It also spits a list of intervals that include an alternation of minimum and maximum points of the curve, which can be used as a musical scale according to the input spectrum.

2 <https://github.com/porres/pd-psycho>.

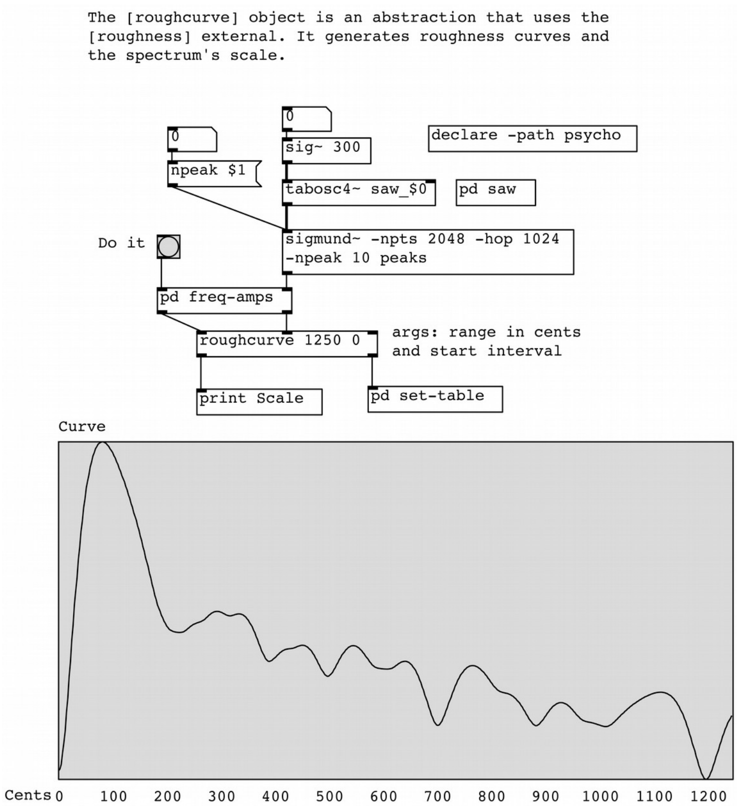


Figure 1: A roughness curve generated with the [roughness] external

3.2 Indigestibility & Harmonicity

These two concepts were developed by Clarence Barlow [11]. The indigestibility of an integer is a measure of how a number may be psychologically 'digestible' according to its prime factors.

Clarence Barlow's harmonicity measure for an interval p/q is defined as the reciprocal of the sum of the indigestibilities of p and q, with its sign indicating the interval's polarity. A negative polarity means that the interval is weighted towards its upper tone and a positive number towards its lower tone. As an example, a perfect fifth has its 'weight' on the lower tone, while a perfect fourth has its weight on the highest tone. The absolute value of a harmonicity is called harmonic intensity.

3.3 Pitch Model objects

The main reference for the objects in this category is the Pitch Model by Parncutt & Strasburger [6] which, on its own, is based on the theoretical work of Ernst Terhardt [12] in Pitch & Consonance, specially Pure/Complex tone Sensation and Sonorousness (which is the same as Tonalness). The main outputs of this Pitch Model are: Tonalness, Pitch Commonality and Pitch Distance.

Tonalness is a measure of Pitch clarity. The Pitch Commonality is a measure of the degree in which two sonorities evoke common pitches. The Pitch Distance is a similar concept that considers the probability of noticing a pitch from one sonority in another, but it's more pertinent in the melodic context.

Just like in the Roughness model, the input is a spectrum defined by lists of frequencies and amplitudes. The frequencies list needs to first be converted to Pitch Category (which is basically a logarithmic MIDI scale). On the other side, the amplitude list needs to first be converted to Pure Tone Audibilities (also defined as the Spectral Pitch Weight), which is a perceptual amplitude measure that takes into account a masking level model.

With this new input of frequencies in Pitch Category and amplitudes in Pure Tone Audibilities, we can recognize a harmonic pattern. The Complex Tone Audibility Ac(P) (or Virtual Pitch Weight) is a degree to which a Harmonic Series (or part of it) is present in the spectrum. For that, a template of 10 harmonics is used, each with a different weight. When there's a match, the Complex Tone Audibility value is registered according to the Pure Tone Audibilities. If a Pure and Complex Tones Audibilities have the same Pitch Category, the greater Audibility value is registered as the Virtual Pitch Weight.

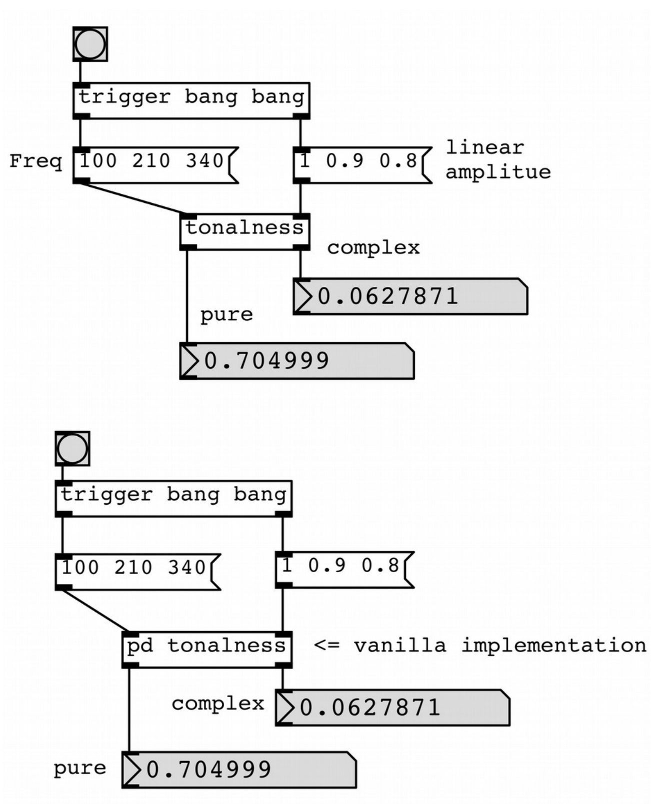


Figure 2: Pure & Complex Tonalness measure with the [tonalness] external and a Pd Vanilla patch

The Pure Tonalness estimation is a normalized quadratic sum of Pure Tone Audibilities Ap(P), and the Complex Tonalness is derived from the maximum Complex Tone Audibility Ac(P). The [tonalness] object outputs both measures. You can set the number of pitch categories. The default is 12 pitches per octave and 10 octaves (hence, an array of 120 values).

Pitch Multiplicity is the number of tones consciously perceived in a sonority. It is calculated from Pitch Audibility A(P), which is given by the maximum value of Pure and Complex Tone Audibilities for each Pitch Category. The Tone Salience is the probability of consciously perceiving a pitch and depends on both Pitch Audibility and Pitch Multiplicity. The [salience] object outputs both the multiplicity value and the Pitch Salience profile, which is the list of Pitch Salience for each Pitch categories (120 by default, like the [tonalness] object).

Successive Pitch Relationships are given by Pitch Commonality and Pitch Distance, both dependent on the Tone Salience output. The Pitch Commonality measure is a Pearson correlation coefficient between two Tone Salience profiles and is given the [commonality] object. The relationship increases according to common Tone Saliences and is equal to 1 in the case of equal spectra and -1 in the case of supposedly complementary ones.

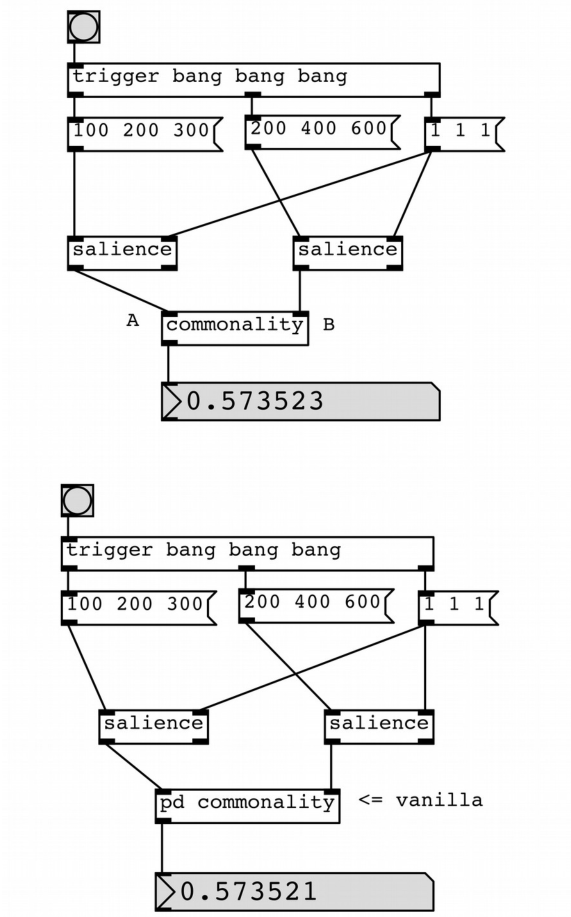


Figure 3: Pitch Commonality measure from Pitch Salience Profiles with the [commonality] external and a Pd Vanilla patch

The Pitch Distance is given by the [distance] object and considers the probability of noticing a pitch from one sonority in another. It takes into account all possible intervals between perceived pitches in two sonorities. The distance is zero for identical sonorities, and exceeds zero otherwise. In the case of Pure Tones, the Pitch Distance is equal to the interval between them in semitones.

3.4 Other objects

This section highlights other relevant objects not presented in the previous sections. Regarding dissonance, the library also contains a [sharpness~] object, which is also part of the Sensory Dissonance Model (and defined as the perceptual equivalent to the spectral centroid). A [centroid~] object is also available as it relates to the measure of sharpness. Also related is the [barks~] object that measures spectral energy per critical band/bark.

Most of the other objects convert frequencies and amplitudes to psychoacoustic scales. For example, for frequencies, we have [hz2bark] and [hz2mel]. For amplitudes, there are objects that convert to Phons, which can be used to plot Equal Loudness Curves.

3.5 Future objects and additions for further releases

Not everything that was developed from the PhD has been fully revised and ported yet, but this is planned for future releases. Currently in the workings, there are a few patches that processes live input. The most important one is a spectral process that alters the spectra relationships of partials to match a given scale. This is known as the spectral mapping technique, also provided in the spectral tools set of objects for Max/MSP [13].

4. Final Considerations and Further work

This paper presented the PSYCHO library for Pure Data, which focuses on psychoacoustic descriptors of Dissonance and is the first library to do so for Pd. It also includes psychoacoustic models never implemented for real time computer music systems, such as the Pitch Commonality model.

This library revisits and restores an earlier research work that had been left virtually abandoned. A drastic makeover provides several advantages such as a friendlier design, a proper documentation, the independence from third party externals and a revision of the code.

The library is also expanding to include audio descriptors as the [centroid~] and [barks~] object. Other objects that are actually low level descriptors are also being considered to be part of the library. For example, raw cepstrum, mel-frequency cepstrum and Bark-frequency cepstrum, which have already been used in Pd for measuring timbral characteristics [14].

References

[1] Porres, A.T. Processos de composição microtonal por meio do modelo de dissonância sensorial, Master dissertation, Unicamp, 2007.

[2] W. Sethares, Tuning, Timbre, Spectrum, Scale (2nd ed.), Springer-Verlag, 2004.

[3] Porres, A.T. “A Roughness Model for An Adaptive Tuning Patch Controlled by Antennas” PdCon07 (2nd International Pure Data Convention) Montreal, 2007.

[4] Porres, A. T. and Pires, A.S. (2009). “Um External de Aspereza para Puredata & MAX/MSP,” in Proceedings of the 12th Brazilian Symposium on Computer Music 2009 (SBCM 2009). Recife, PE: Braz

[5] Porres, A. T. Modelos psicoacústicos de dissonância para eletrônica ao vivo, PhD thesis, USP, 2012.

[6] R. Parncutt, H. Strasburger, “Applying psychoacoustics in composition: “Harmonic" progressions of "non-harmonic" sonorities”, Perspectives of New Music, 1994, 32 (2), pp. 1-42.

[7] Porres, A.T. “Dissonance Model Toolbox in Pure Data” Proceedings of the 4th Pure Data International Convention (PdCon11), Berlin, Germany, 2011.

[8] H. Helmholtz, On the Sensations of Tone as a Psychological basis for the Theory of Music (2nd ed.), Dover Publications, 1954.

[9] C. Barlow, “Von der Musiquantenlehre” Feedback Paper 34, 2008. Feedback Studio Verlag.

[10] P.N. Vassilakis. “Perceptual and Physical Properties of Amplitude Fluctuation and their Musical Significance.” Ph.D. Thesis, 2001, UCLA.

[11] Barlow, Clarence & Lohner, Henning. (1988). Two Essays on Theory. Computer Music Journal. 11. 44.

[12] E. Terhardt, “The concept of musical consonance: A link between music and psychoacoustics”, Music Perception, 1984, Vol.1, pp. 276-295.

[13] W. A. Sethares, A. J. Milne, S. Tiedje, “Spectral Tools for Dynamic Tonality and Audio Morphing” Computer Music Journal, 2009, Volume 33, Number 2, Summer 2009, pp. 71-84.

[14] Brent, William. Perceptually Based Pitch Scales in Cepstral Techniques for Percussive Timbre Identification: a comparison of raw cepstrum, mel-frequency cepstrum, and Bark-frequency cepstrum. From the proceedings of the ICMC in Montreal, 2009.

Low-Latency f0 Estimation for the Finger Plucked Electric Bass Guitar Using the Absolute Difference Function

Christhian Fonseca [1,2], Tiago Tavares[1,2]

[1]School of Electric and Computer Engineering (FEEC) – University of Campinas (Unicamp)
CEP 13083-852 – Av. Albert Einstein, 400 – Campinas – SP – Brazil

[2]Interdisciplinary Nucleus for Sound Studies (NICS – University of Campinas (Unicamp)
CEP 13083-872 – Rua da Reitoria, 165 – Campinas – SP – Brazil

christian_h@hotmail.com

Abstract. *Audio-to-MIDI conversion can be used to allow digital musical control by means of an analog instrument. Audio-to-MIDI converters rely on fundamental frequency estimators that are frequently restricted to a minimum delay of two fundamental periods. This delay is perceptible for the case of bass notes. In this paper, we propose a low-latency fundamental frequency estimation method that relies on specific characteristics of the electric bass guitar. By means of physical modelling and signal acquisition, we show that the assumptions of the method relies on generalize throughout electric basses. We evaluate our method in a dataset with musical notes played by diverse bassists. Results show that our method outperforms the Yin method in low-latency settings, which indicates its suitability for low-latency audio-to-MIDI conversion of the electric bass sound.*

1 Introduction

Digital instruments and controllers commonly use the MIDI (Musical Interface for Digital Instruments) standard to communicate with each other. This allows combining different synthesizers, controllers, and effect racks, which expands the expressive possibilities related to timbres, musical performances, musical recordings and notations [1]. This toolchain can use analog instruments by means of audio-to-MIDI converters [2].

Audio-to-MIDI converters aim at identifying the notes played by the instrument. For such, they use a perceptual model in that relates the fundamental frequency (f0) of an audio signal to its pitch. There are many known algorithms that aim at estimating f0, such as the autocorrelation [3] and the Yin method [4].

f0 estimators commonly aim at finding periodicity in an audio signal f_j. The periodicity is based on the model

$$f_j = f_{j+kJ}, \tag{1}$$

where J is the fundamental period of f_j and $k \in \mathbb{Z}$. Methods that rely on this property commonly require analyzing at least two fundamental periods of the signal. This incurs in a lower-bound for the latency of Audio-to-MIDI conversion that can be close to 50 ms for the lowest notes (41.2 Hz) in standard 4-string electric basses. These long delays can harm the use of basses as a MIDI controller.

In this work, we aimed at attenuating this problem using an f0 estimation method especially crafted for the electric bass. The method exploits specific properties of the electric bass waveform. Our method allows f0 estimation with an algorithmic latency of 1.1 times the fundamental period of the signal.

Experimental results show that the method is effective with an error rate of 15%. This is half of error rate of the baseline method (Yin).

2 Related work

Pitch is an auditory sensation often related to the perception of a repetition rate of a waveform [5]. The repetition rate is called Fundamental Frequency (f0) and can be used to decompose harmonic complex tones into sinusoidal harmonic components whose frequencies at multiple integers of the fundamental frequency f_0, that is:

$$f(t) = \sum_{m=1}^{M} a_m \cos(2\pi m f_0 t + \phi_m). \tag{2}$$

The relative harmonic amplitudes a_m are commonly associated to timbre differences and the fundamental frequency f_0 is closely related to the sensation of pitch [6]. In this study, we assume that the fundamental frequency is the physical counterpart of the sensation of pitch, hence estimating the fundamental frequency is equivalent to finding the pitch of a signal.

There are several methods that aim at finding the pitch of periodic signals, as discussed next.

2.1 Autocorrelation

A common method for estimating pitch of periodic signals is by detecting the greatest positive peak of the autocorrelation function r_t [3], which is calculated by:

$$r_t(\tau) = \sum_{j=t+1}^{t+W} f_j f_{j+\tau} \tag{3}$$

The autocorrelation $r_t(\tau)$ is a measure of the similarity between the signal f_j and a temporally shifted version $f_{f+\tau}$ of itself. It presents peaks in values of τ that correspond to the fundamental periods of f_j.

2.2 Harmonic Sum Spectrum

The Harmonic Sum Spectrum is obtained by adding the Fourier transform of the signal is to sub-sampled versions of itself [7]. This corresponds to calculating:

$$Y(\omega) = \sum_{m=1}^{M} F(m\omega), \tag{4}$$

where F is the frequency spectrum of f_j, Y is the Harmonic Sum Spectrum and M is the number of downsampling processes to execute. After this processing, the argument of the global maximum of Y corresponds to the f0.

2.3 Yin method

The Yin method was proposed by Cheveigné and Kawahara [4]. It is based on the same premises as the autocorrelation method, with the addition of a series of modifications that reduce errors. Two very important modifications are the substitution of the autocorrelation function by the difference function shown in Equation 5, and the application of the a cumulative mean normalized difference function shown in Equation 6.

$$d_t(\tau) = \sum_{j=1}^{W} (f_j - f_{j+\tau})^2. \tag{5}$$

$$d_t^{'}(\tau) = \left\{ \begin{array}{ll} 1 & \text{, if } \tau = 0 \\ \frac{d_t(\tau)}{(1/\tau)\sum_{j=1}^{\tau} d_t(j)} & \text{, otherwise} \end{array} \right\} \tag{6}$$

The shortest period between the local minima of $d_t^{'}$ that are lower than a pre-defined threshold is yielded as the fundamental period of f_j.

2.4 Discussion

All the methods discussed in this section directly rely on the periodicity property as stated in Equation 1 or the harmonic series model shown in Equation 2. This allows them to be applicable for the general case of finding pitch in periodic signals, but bounds them to a minimum delay of twice the fundamental period.

In this work, we propose a pitch detection method that relies on specific characteristics of the plucked electric bass string. This restricts our method to signals generated by this specific instrument. However, it allows reducing the delay to 1.1 times the fundamental period.

This reduction is critical for the real-time pitch detection in low-pitch notes. In these notes, general-purpose methods require a delay of around 50ms to work properly. Our method allows detecting the same pitch with a delay of around 30ms.

The method proposed by [2] also indicates to estimate f0 close to the theoretical minimum latency, i.e. the fundamental period of the lowest observable pitch, but with higher computational complexity.

The proposed method is based on specific properties of the plucked electric bass signal. These properties are analyzed using a physical model, which guide its generalization possibilities. Then, the proposed model is compared to the Yin method using a dataset containing recordings from electric bass guitars.

3 Time-domain Behavior of a Plucked String

This section discusses the properties of the plucked string signal that were used as basis for our f0 estimation method. These properties were inferred by analyzing the audio signal of an electric bass strings, as shown in Section 3.1, then the physical model discussed in Section 3.2 was used to generalize these results, as shown in Section 3.3.

3.1 Plucking an Electric Bass String

The traditional electric bass guitar is an electro-acoustic instrument with a body and neck made of wood and four metal string tuned to E, A, G and D, which are fixed in a metal bridge on the body and in the nuts of the neck. The neck has a fingerboard with 20 to 24 frets which divides it in tonal areas. The index and middle fingers of the right hand are used to pluck the strings and the fingertips of the left hand are used to hold the strings against the fretted fingerboard. This changes the free length of the string, which modulates its natural oscillation frequency.

There are magnetic pickups placed on the instruments body, under the strings. They convert the string transverse velocity at its position into an electric voltage. The string transverse velocity can be see as a wave which propagates from the pluck position along the string length, reflecting and inverting when reach the string end, as shown in Figure 1.

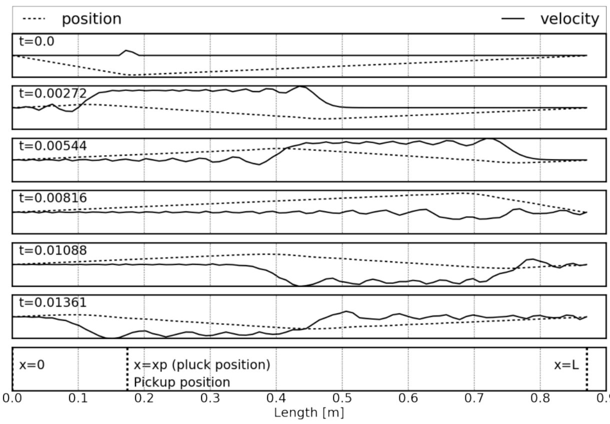


Figure 1: Position and velocity of the string along the x axis at different times t

The waveforms of the voltage signal at the pickups, as shown in Figure 2, indicates repetitions of a peak at the beginning of each cycle. In order to confirm that this characteristic is maintained for all electric bass guitars (instead of being a characteristic of the specific instrument), the behavior of its string was mathematically modeled, as discussed in the next section.

3.2 Physical model

The behavior of the bass string can be modelled using an ideal string along the coordinate x with fixed ends at $x = 0$ and $x = L$, which give us the following boundary conditions:

$$y(x = 0, t) = 0. \quad (7)$$

$$y(x = L, t) = 0. \quad (8)$$

The string has linear density μ and is stretched with a force T. It is initially at rest and is plucked in the position $x = x_p$ with amplitude $y(x_p, 0) = A$. In this situation, the initial transverse displacement $y(x, 0)$ can be expressed by

$$y(x, t = 0) = \left\{ \begin{array}{cc} A(\frac{x}{x_p}) & \text{, if } x < x_p \\ A(1 - \frac{x - x_p}{L - x_p}) & \text{, otherwise} \end{array} \right\} \quad (9)$$

and the velocity distribution $y'(0, x)$ is

$$y'(x, t = 0) = 0. \quad (10)$$

For a short segment of this string between x and Δx there is a slope $\delta y/\delta x = \tan(\theta)$ and a vertical force F defined by:

$$F = T \sin(\theta)(x + \Delta x) - T \sin(\theta(x)) \quad (11)$$

If y corresponds to a small displacement, θ is also small and can be approximated using $\cos(\theta) \approx 1$ and $\sin(\theta) \approx \tan(\theta)$. This allows re-writing Equation (11) as:

$$F = T(\frac{\partial y}{\partial x}(x + \Delta x) - \frac{\partial y}{\partial x}(x)) \quad (12)$$

Using the Newton's second law:

$$F = m\frac{\partial^2 y}{\partial t^2} \quad (13)$$

and knowing that the mass for this string segment is $m = \mu \Delta x$, we have:

$$T(\frac{\partial y}{\partial x}(x + \Delta x) - \frac{\partial y}{\partial x}(x)) = \mu \Delta x \frac{\partial^2 y}{\partial t^2} \quad (14)$$

dividing both sides of Equation (14) by Δx, and making $c = \sqrt{T/\mu}$, it becomes the wave equation:

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2}, \qquad x \in (0, L), t \in (0, T] \quad (15)$$

This model was used to simulate plucked strings and the resulting waveforms were compared to measured waveforms, as discussed in Section 3.3.

3.3 Plucked string simulation

Equation 15 was numerically solved using the finite difference method [8] and the algorithmic steps used by Langtangen [9]. The Taylor series expansion was used to approximate it as:

$$\begin{aligned} &\frac{y(x + \partial x, t) - 2y(x, t) + y(x - \partial x, t)}{\partial x^2} = \\ &\frac{1}{c^2} \frac{y(x, t + \partial t) - 2y(x, t) + y(x, t + \partial t)}{\partial x^2} \end{aligned} \quad (16)$$

Using the i, j notation such that $y(x, t) = y_{ij}$, inserting the wave number $C = \frac{c\partial t}{\partial x}$ and rearranging Equation 16 yields:

$$y_{i,j+1} = C^2 y_{i-1,j} + 2(1 - C^2) y_{i,j} + C^2 y_{i+1,j} - y_{i,j-1}. \quad (17)$$

To calculate the value of this function in the first time step, $y_{i,j-1}$ must be determined. This can done using the initial velocity in Equation 10 and Tailor's series as follows:

$$\frac{y(x, t + \partial t) - y(x, t - \partial t)}{2\partial t} = 0. \quad (18)$$

Rearranging equation 18 and rewriting in the i, j notation, we find that:

$$y_{i,j-1} = y_{i,j+1}. \quad (19)$$

Finally, replacing $y_{i,j-1}$ by $y_{i,j+1}$ in Equation 17, isolating $y_{i,j-1}$ and dividing both sides by 2, we have:

$$y_{i,j+1} = \frac{C^2}{2} y_{i-1,j} + (1 - C^2) y_{i,j} + \frac{C^2}{2} y_{i+1,j}, \quad (20)$$

which is the finite difference scheme. The numerical simulation was executed over the discrete spatial domain [0,L] equally spaced by ∂x and over the discrete temporal domain $[0, T]$ equally spaced by ∂t.

The model's pluck position $x_p = L/5$ and the string length $L = 0.87m$ were directly measured from the strings of an electric bass. The wave velocity c was calculated using $c = f/(2L)$ [10] related to note E0. The simulation time was define as $T = 0.05s$.

Over the spatial domain, the algorithm computes $y_{i,0}$ using Equation 9 and $y_{i,1}$ using Equation 20 and applying the boundary conditions from Equations 7 and 8. Then, for each element j from temporal domain, apply Equation 17 to find $y_{i,j+1}$ for each element i from the spatial domain, applying the boundary conditions again.

The output simulated signal was retrieved from the string velocity in the position $x = L/5$, approximately the pick up position and was yielded to a 5th order low-pass Butterworth filter with a $150Hz$ cutoff frequency. This simulates the smoother bend of the string due to its stiffness and the soft touch from the fingertip, which are responsible for generating tones with weaker high-frequency

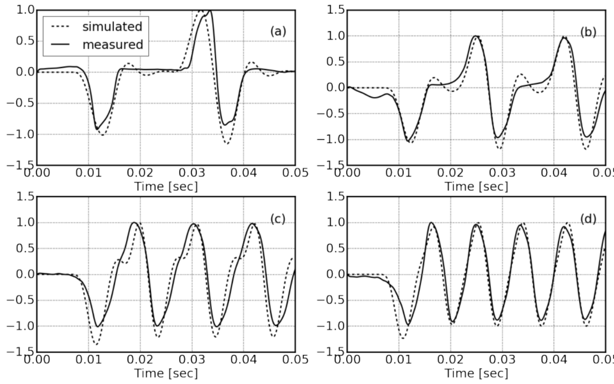


Figure 2: Simulated and measured notes played on string E of the electric bass guitar (a) E0 (b) A♯0 (c) F1 (d) A1

components [11]. The resulting signals were compared to the recorded signals, as shown in Figure 2.

Figure 2 shows that the physical model generates shapes that are similar to those found in the acquired signals. This means that the peak behavior is not a particular behavior of the specific electric basses that were used in our acquisitions. Rather, this behavior can be expected to appear in electric basses in general, hence it can be used for further steps in fundamental frequency estimation.

4 Fundamental Frequency Estimation

The simulated and measured waveforms in Figure 2 show that there is a peak at the onset of the note and in the beginning of each cycle after it. These peaks have approximately the same width, regardless of the note's frequency, and the note's fundamental frequency occurs due to the rate in which peaks appear in the signal. The proposed method is based on these two characteristics, as follows.

The proposed method consists of the application of the signal to an absolute difference function:

$$d_t(\tau) = \sum_{j=1}^{W} |f_j - f_{j+\tau}|, \tag{21}$$

with a window size W shorter than half width of this first peak as shown in figure 4(a). This thin window plays an important role to make possible for the method find f0 after 1.1 times the fundamental period, whereas the Yin method needs more than two fundamental periods [4], as shown in Figure 3.

The f0 can be estimated from the interval between the two bigger dips in the absolute difference function, which are illustrated in Figure 4(b). In order to reduce error rates, the absolute difference function was normalized to the $[0, 1]$ range and a maximum threshold was applied. This allows ignoring the dips whose absolute value is too high.

If the specific string being played is known *a priori*, minimum and maximum frequencies can be used to reduce errors. They allow ignoring the dips that lead to frequency estimations out of this range.

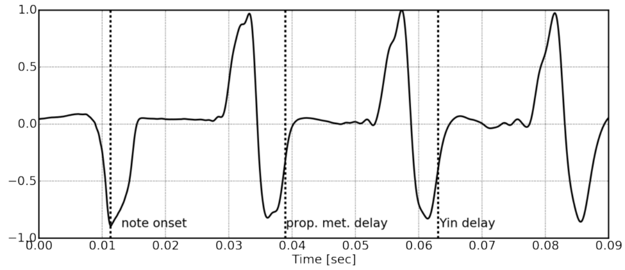


Figure 3: Algorithmic delay for the proposed method and for the Yin method.

Also, for frequencies higher than twice the minimum frequency parameter, there are more than two dips. In this case, the fundamental period is estimated from the average of the intervals between the dips of the difference absolute function, as the illustrated in Figure 4(d).

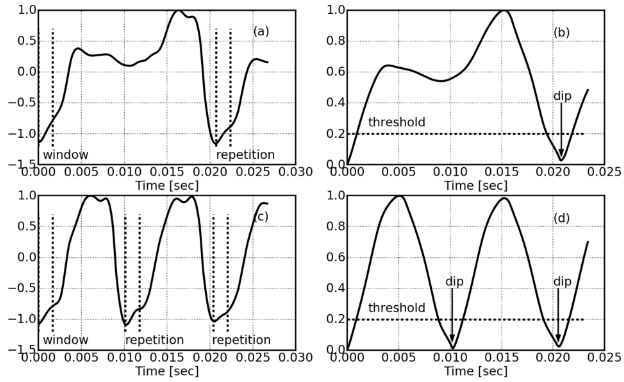


Figure 4: Sample analyzed $f(t)$ and window size W for the E string notes: (a)$G0$ and (c)$G1$. Absolute difference function $d(\tau)$ and threshold: (b)$G0$ and (d)$G1$.

The next section discusses experiments regarding the proposed method.

5 Experiments and results

5.1 Dataset

The proposed method was tested using a set of audio recordings acquired from 3 different electric bass guitars. Each of them was played by a different musician, and all of them used the finger-plucking technique. All notes within the instrument's range were recorded from each of the guitars, using two different instrument equalizations (full bass and full treble). This yielded 528 recordings, which were all manually cropped to start at the note onset.

5.2 Experiments

This section describes experiments that compare the proposed method to the Yin method [4], as implemented by Guyot [12]. The experiments comprised executing both the proposed method and the Yin method to estimate the f0 in the dataset samples.

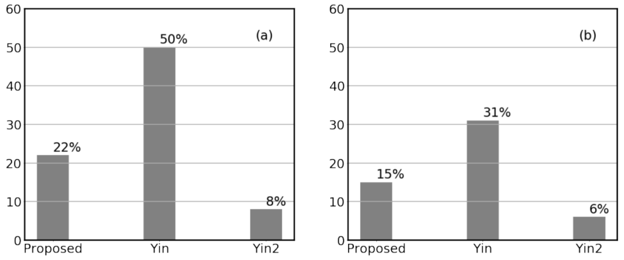


Figure 5: (a) Test 1 error rates . (b) Test 2 error rates

5.2.1 Test 1 - sample length for note

In this first test, the sample length provided as input parameters for the algorithms are equal to $1.1 \times T_{t1} \times fs$, being T_{t1} the fundamental period of the expected note and fs the frequency of sampling of the digital audio signal. To serve as reference, the test was repeated for the Yin method with a sample length equal to $2.1 \times T_{t1} \times fs$ and is referenced as ”Yin2” in figure 5 (a).

5.2.2 Test 2 - sample length for string

This second test is a more common application for a pitch detector in a string instrument, where the fundamental frequency should be estimated from a range of approximately 2 octaves. So, the sample length provided as input parameters for the algorithms are equal to $1.1 \times T_{t2} \times fs$, being T_{t2} the fundamental period of the lower note from the specific string to which the recorded note belong. Also in this case, the test was repeated for the Yin method with a sample length equal to $2.1 \times T_{t2} \times fs$ and is referenced as ”Yin2” in figure 5 (b).

To determined if the method fails, the MIDI note correspondent to the fundamental frequency estimated is calculated as:

$$M_{note} = 12 \log \left(\frac{f_0}{16.351597} \right) \frac{1}{\log(2)} + 0.5, \qquad (22)$$

being f_0 the estimated fundamental frequency, 16.351597 the f_0 for the MIDI note = 0 and 0.5 as tolerance, as the result will be truncated. If the calculated MIDI note differs from the expected one, it is counted as one error.

5.3 Discussion

The error rates presented in Figure 5 show that the proposed method had less than half of Yin method's error rate, so having a better performance estimating f0 on both tests. As expected, the Yin method is a better solution when sample length is longer than 2 cycles of the fundamental period, but for the string E of a electric bass guitar, only the algorithmic delay should be higher than 50 ms ($2/f0 = 2, 1/41.20Hz \approx 0, 051s$), which is perceptible for a bass player, making it harder to play a bass guitar with real-time MIDI outputs.

The next section shows conclusive remarks.

6 Conclusion

A method based on the absolute difference function and on the waveforms from a finger plucked strings of a electric bass guitar was presented. It was tested over 528 notes recorded from three different bass guitars and it shows to be capable to estimate these notes from samples with length equal to 1.1 times their fundamental periods, while our reference method, Yin, under the same conditions, had double the error rate. This shorter algorithmic delay, near the minimal theoretical delay (one fundamental period) and low computational complexity, makes the proposed method suitable for real time applications for the electric bass guitar, such as a MIDI bass guitar.

However the method missed 15% of the notes on test 2, which is a similar application, so future studies should be made to improve this results. Also, new recordings in which the bass players always pluck the string smoothly in order to keep the first cycles of the signal similar to the modeled ones, can show an alternative way to a MIDI bass guitar. This imposes a limited way to play in exchange for a more precise note detection and lower latency. Lastly, the method was not tested for notes played on a vibrating string which certainly should make harder to estimate the correct f0. This case will be approached in future work.

7 Acknowledgements

Thanks are due to Mario Junior Patreze from ”Escola de Música de Piracicaba Maestro Ernst Mahle”, Marcio H. Goldschmidt and Giovani Guerra from musical studio ”Esgoto” for the records from their own electric bass guitars which compose the database of this work. This study was financed in part by the ”Coordenação de Aperfeiçoamento de Pessoal de Nível Superior” - Brasil (CAPES) - Finance Code 001.

References

[1] Warren A. Gibson, J. *The MIDI Standard.* http://www.indiana.edu/ emusic/361/midi.htm, accessed 05/9/2019.

[2] O. Derrien. A very low latency pitch tracker for audio to midi conversion. *17th International Conference on Digital Audio Effects (DAFx-14)*, 2014.

[3] L. Rabiner. On the use of autocorrelation analysis for pitch detection. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(1):24–33, February 1977.

[4] Alain de Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[5] E. J. Heller. *Why You Hear What You Hear.* Princeton University Press, 2012. (Chapter 23; pp. 437-504).

[6] Andrew J. Oxenham. Pitch perception. *Journal of Neuroscience*, 32(39):13335–13338, 26 September 2012.

[7] A. M. NOLL. Pitch determination of human speech by the harmonic product spectrum, the harmonic surn spectrum, and a maximum likelihood estimate. *Symposium on Computer Processing in Communication, ed.*, 19:779–797, 1970.

[8] M.K. Jain. *Numerical Methods for Scientific and Engineering Computation*. New Age International, 1st ed. edition, 2003. ISBN-10: 8122414613. pp: 844.

[9] H.P. Langtangen. *Finite difference methods for wave motion*. Department of Informatics, University of Oslo, preliminary version edition, 2016.

[10] F. Iazzetta. *Tutoriais de Audio e Acustica*. http://www2.eca.usp.br/prof/iazzetta/tutor/acustica, accessed 04/25/2019.

[11] E. Jansson. *Acoustics for Violin and Guitar Makers*. http://www.speech.kth.se/music/acviguit4/, 4th ed. edition, 2002. (Chapter 4; pp. 16-18).

[12] Guyot P. Fast python implementation of the yin algorithm. *http://doi.org/10.5281/zenodo.1220947*, 2018. accessed 01/02/2018.

Comparing Meta-Classifiers for Automatic Music Genre Classification

Vitor Yudi Shinohara [1]*, Juliano Henrique Foleiss[1] , Tiago Fernandes Tavares[1]

[1] School of Electrical and Computer Engineering - University of Campinas - Brazil
Albert Einstein, 400 – 13083-852, Campinas, SP

vitorys@dca.fee.unicamp.br, julianofoleiss@utfpr.edu.br, tavares@dca.fee.unicamp.br

Abstract. *Automatic music genre classification is the problem of associating mutually-exclusive labels to audio tracks. This process fosters the organization of collections and facilitates searching and marketing music. One approach for automatic music genre classification is to use diverse vector representations for each track, and then classify them individually. After that, a majority voting system can be used to infer a single label to the whole track. In this work, we evaluated the impact of changing the majority voting system to a meta-classifier. The classification results with the meta-classifier showed statistically significant improvements when related to the majority-voting classifier. This indicates that the higher-level information used by the meta-classifier might be relevant for automatic music genre classification.*

1 Introduction

Music genres are categories that group songs with same characteristics, such used instruments, rhythm and music's harmony [1]. It can be used to organize music track collections [2]. Automatic Music Genre Classification (AMGC) is a Music Information Retrieval task that aims at facilitating the labeling of tracks according to their genre [1].

AMGC relies on representing music tracks in a vector space using sound-related features. Some approaches use a single vector to represent each track, whereas others use multiple vectors for this representation. A single-vector representation (SVR) is more compact, but can result in loss of information due to long-term summarization [3]. Conversely, the multiple-vector representation (MVR) depends on an additional step for combining the information derived from each of the vectors [4]. Each vector in the MVR representation represents different sounds from the track, leading to a richer representation.

A possible approach for combining information from MVR is using voting mechanisms, in particular, majority voting [4, 5]. This method relies on classifying each frame individually and then selecting the most frequent label as the one associated to the track. It relies on the idea that classification errors are less frequent than correct classifications, hence having multiple attempts for each track reduces the probability of an overall error.

In this paper, we explore diverse methods for combining texture classifications into a final prediction per track. For such, we use the outputs of a texture genre classifier as the input for a meta-classification stage. This relies on the hypothesis that some classification errors are typical in particular genres, thus these errors can be exploited to improve classification results.

We evaluated three different approaches for combining the classifications in the meta-classification stage. We considered the case where the output for each texture is just the predicted class, not the estimated probability for each class. The baseline is the widely used majority voting [6, 7, 8, 9, 10]. We evaluated meta-classifiers based on two different representations. In the first one, the classification histogram is used as a feature vector, which is yielded to the meta-classifier. Second, we used the sequence of texture class predictions as inputs to time-series classifiers.

The approaches were evaluated in four different datasets. Our results indicate that majority voting is an effective technique for datasets containing full-length popular music tracks, and time-series classification is more accurate when textures are typically more uniform throughout the track. This suggests that the changes in musical textures throughout a track can be relevant for genre classification, but they can be hard to model using general-purpose tools in heterogeneous tracks.

This paper is structured as follows: In Section 2 we introduce the meta-classifier architecture, features and data sets used in the evaluation. In Section 3, meta-classifier results are compared with majority voting results. In Section 4, final considerations are presented.

2 Method

The evaluation system is composed of two stages: texture classification and meta-classification. The first stage outputs a class prediction for each texture of an input music track. In the second stage, the textures of a music track are used to build a track representation that is input into a meta-classifier, which yields a final classification for the track.

2.1 Music Texture Classifier System

The Music Texture Classifier System (MTCS) outputs a class prediction for each texture of an input music track. A texture is a feature vector aggregated over a sequence of feature vectors calculated over audio frames. Textures aim to encode audio content of a relatively long (typically 1s to 5s) audio segment, which is useful for genre classification [1]. Figure 1 shows the MTCS architecture. Given a music track sampled at 44Khz, a 2048-sample Short-Time Fourier Transform (STFT) is calculated, with 50% overlap. This yields 23ms frames. Then, a set of hand-crafted features are calculated for each frame. This feature set consists of the following features: Spectral Cen-

*Supported by CAPES.

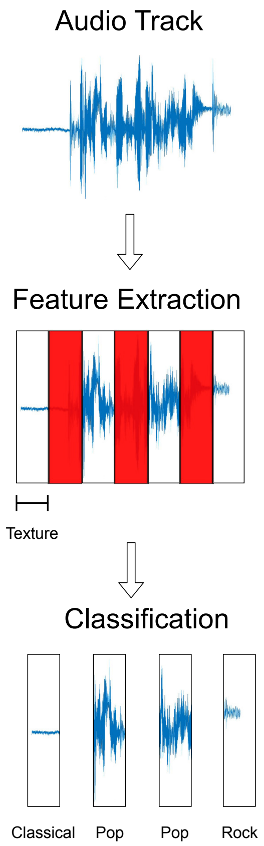


Figure 1: Music Texture Classifier System (MTCS) architecture. It receives an audio file as input and yields a sequence of label predictions related to each segment of the track.

troid, Spectral Rolloff, Spectral Flux, Energy, Zero Crossing Rate [1], Spectral Flatness [11], and the first 20 MFCC coefficients [12]. The feature vector also consists of the first and second-order derivatives of each feature. Thus, each frame-level feature vector has 78 features.

Textures are calculated using the mean and variance of each feature in every 10 low-level frames, resulting in a 10x downsample. This yields a sequence of 156-dimensional feature vectors.

However, the total number of textures in the training set can become too large. To make training tractable, the texture set for each track is further downsampled by selecting k linearly-spaced textures. Since k is a parameter, it was evaluated as 5, 20 and 40 in our experiments.

Along with the votes for each texture of a track, the MTCS also yields a final label for each track. This label is computed by a voting procedure, in which the class with the most votes is decided as the track label. These results are used as baseline.

2.2 Meta Classification

This paper explores how the MTCS votes can be combined via meta-classifiers. The votes were organized in two different representation as input for the meta-classifier: vote histograms and sequences of votes. Histograms represents the number of votes for all genres in a specific track. Sequences of votes indicate the vote progression along the track. In other words, histograms disregard the order of the textures, whereas sequences of votes rely on this information.

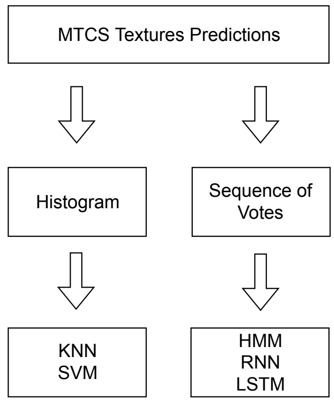


Figure 2: Histogram and sequence of votes classification scheme.

2.2.1 Histogram Classification

The left-hand side of Figure 2 shows how histograms are used in our meta-classification approach. First, texture votes are obtained from the MTCS for every track in the dataset. Then, vote histograms are built by computing the number of votes each class received. These histograms are used to describe music tracks. The true track labels were used as target values. Two classifiers were evaluated as meta-classifiers, The K-Nearest Neighbor (KNN) and the Support Vector Machine (SVM).

The systems were evaluated using K-fold cross-validation. To make it easier to compare to the baseline, the same folds were used to evaluate the histogram meta-classifiers. Thus, the same training sets were used for training the meta-classifiers, while the same testing sets were used for evaluating them.

For hyper-parameter tuning, the training set was randomly split into a training set (80%) and a validation set (20%). A grid-search was used for hyper-parameter tuning. The classifier parameters evaluated are shown in Table 1.

Table 1: Hyper-parameters evaluated with grid-search for histogram classifiers.

Classifier	Hyper-parameter	Range
SVM	C	{ 0.001, 0.01, 0.1, 1, 10, 100 }
SVM	*Gamma*	{ 0.001, 0.01, 0.1, 1 }
KNN	k	{ 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 }

2.2.2 Sequence of Votes Classification

The right-hand side of Figure 2 shows how the sequences of votes are used for meta-classification. The MTCS outputs the votes following the original sequence of the corresponding textures in the track. The sequences are then used as input to the meta-classifier. Three sequence classifiers were evaluated with the sequences as input. Hidden Markov Models (HMM) [13] can be used as classifiers. A HMM is trained for each target class. Then, test sequences are evaluated, and each HMM predicts the probability it generated the sequence. The target class corresponding to the maximum probability HMM is assigned to the sequence. Grid-search was used to find the best hyper-parameter combination on the validation set. Table 2 presents the hyperparameter evaluation values used in our experiments.

Table 2: HMM Hyper-parameters tuned with grid-search.

Hyper-Param	Tested Values
Number of Hidden States	{3, 5, 7, 9}
Covariance Matrix	{full, diag}
Number of Iterations	{50, 100, 150}

We also evaluated Recurrent Neural Networks as meta-classifiers. Table 3 presents the architecture used in our experiments, along with the evaluated parameter values. Two types of recurrent cells were evaluated: simple Recurrent Neural Network (RNN) and Long ShortTerm Memory (LSTM) [14]. The Long Short Term Memory mitigates the vanishing gradient problem that occurs in traditional RNNs. This problem is known to get worse as the sequences get longer.

Table 3: Neural network architecture used in the experiments.

Layer	Type	Activation Function	Neurons
1	Dense	Linear	1
2	RNN / LSTM	tanh	{20, 30, 40, 50}
3	Dense	ReLu	{5, 10, 15}
4	Dense	Softmax	{9, 10, 13}

2.3 Datasets

The datasets used in the experiments are presented in Table 4. These datasets are widely used by the MIR community and are publicly available. These datasets vary greatly in terms of music content, label balancing, number of tracks and track length.

Table 4: Description of evaluation datasets.

Data set	# Tracks	# Genres	Balanced	Track Len.	Folds
GTZAN	1000	10	Yes	30 s	10
LMD	1300	10	Yes	Full	3
HOMBURG	1886	9	No	10 s	10
EXBALLROOM	4180	13	No	30 s	10

The GTZAN dataset [1] is one of the most widely used datasets in genre recognition research [15]. It consists of 10 western genres, which greatly vary with respect to spectral patterns. The folds were created randomly. The HOMBURG dataset [16] presents a challenge for systems based on texture classification, since the tracks are only 10s long. The Extended Ballroom dataset (EXBALLROOM) [17] is also challenging for texture classification systems. Because this dataset is made of ballroom dances, there are subsets of genres that use the same instrumentation. Thus, time-related features, such as rhythm and tempo descriptions, are needed in order to achieve good results. An artist filter [18] was applied to EXBALLROOM during fold splitting. A subset of the Latin Music Database (LMD) [19] was also used in the evaluation. This is the only dataset evaluated that consists of full-length tracks. A subset of the original dataset was used for artist filtering, since some genres were largely represented by only a few artists.

In the next section we present the evaluation results.

3 Results

In this section we present the classification accuracy for all the meta-classification approaches presented in this paper. The results are the average and standard deviation across all folds. The number of folds varies depending on the dataset and are shown in Table 4. Statistical significance was evaluated by the Student's T-test. Statistical significance was evaluated for all meta-classification approaches when compared to the Majority Vote baseline. A threshold of 5% was considered for rejecting the null-hypothesis.

Table 5 shows the best results for each classifier in the four datasets evaluated. Statistically significant results are shown in bold.

The results from KNN and SVM were similar. For all results, the difference is not statistically significant. KNN is known to be able to perform well in low-dimensional data. As the largest feature vector had 40 features, the dimensionality did not have a big impact on the KNN results. Furthermore, KNN has a lower training computational cost. Thus, in the evaluated datasets, KNN offers a superior cost/benefit ratio. However, only the SVM was statistically superior than the majority voting baseline.

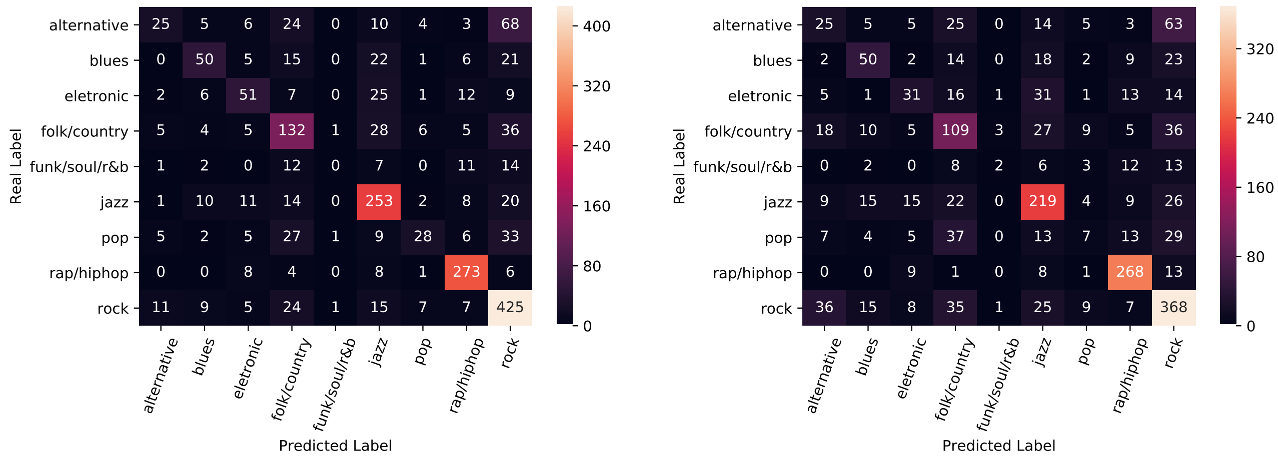
Both neural network results were not statistically superior to the histogram results. This suggests that for the datasets evaluated, the sequence of the votes is not key for performance improvement. Similarly to the SVM, both the RNN and LSTM networks performed better than the baseline.

The confusion matrix for the RNN metta-classifier and the HOMBURG dataset is shown in Figure 3. This was the best average result obtained overall for this dataset ($65\% \pm 0.08$). Figure 3 shows the confusion matrix for the majority voting baseline.

Overall, the results of the histogram-based approaches were similar compared to the majority vote. The majority voting structure is embedded in cases where the histogram was correctly classified by majority vote. When presenting a histogram whose majority vote is correct, the classifier tends to associate the behavior of the majority vote. Therefore, the majority vote seems to represent a lower limit for histogram classification in the evaluated datasets.

Table 5: Best results for all meta-classifiers.

	Baseline	Proposed Method				
		Histograms		Sequence of Votes		
	Majority Vote	KNN	SVM	HMM	RNN	LSTM
GTZAN	0.79 ± 0.06	0.75 ± 0.16	0.76 ± 0.15	0.53 ± 0.04	0.73 ± 0.14	0.79 ± 0.16
LMD	0.82 ± 0.02	0.81 ± 0.03	0.82 ± 0.02	0.69 ± 0.01	0.72 ± 0.07	0.75 ± 0.07
HOMBURG	0.54 ± 0.03	0.58 ± 0.05	$\mathbf{0.59 \pm 0.04}$	0.53 ± 0.02	$\mathbf{0.65 \pm 0.08}$	$\mathbf{0.64 \pm 0.07}$
EXBALLROOM	0.75 ± 0.02	0.77 ± 0.05	0.77 ± 0.05	0.62 ± 0.05	0.76 ± 0.04	0.78 ± 0.04



(a) RNN meta-classifier.

(b) Majority Voting baseline.

Figure 3: Confusion matrices for the RNN meta-classifier and the baseline systems in the HOMBURG dataset.

4 Conclusion

Various research on Automatic Music Genre Classification use the multiple-vector representation to describe tracks. When only the votes, no probabilities, are available for each texture, a final track classification is decided by majority voting. This paper presented two alternative approaches for combining texture votes into genre predictions. The first method evaluated builds a vote histogram. This histogram is used to represent the track for the meta-classifier, maps histograms into final genre decisions. The second method relies on the sequences of votes for each track. The sequences are then input into sequence classifiers, which map sequences of votes into genres.

The histogram classifiers tends to have a better performance when music textures are more uniform through time. In contrast, the meta-classifiers based on sequences of votes obtain better results on data sets in which musical textures are more heterogeneous. This suggests that LSTM and RNN architectures were effective in modelling the short-term (close to 10s) sound changes that characterize textures, but were unable to derive differences related to musical structure.

Therefore, recurrent neural networks are effective model dependencies on short, few-seconds scale, whereas long-term dependencies should be investigated using other models. The exploration of model behaviors for both of these time scales is an interesting venue for future work.

In conclusion, the source code from the proposed meta-classifiers is available at a public GitHub repository[1]. Instructions for experiment execution and datasets used are also included.

5 Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior - Brasil (CAPES) - Finance Code 001.

References

[1] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[2] CBS Interactive. Last.fm, 2002. Accessed: 2019-05-01.

[3] J. Lee and J. Nam. Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging. *IEEE Signal Processing Letters*, 24(8):1208–1212, Aug 2017.

[4] Tiago Fernandes Tavares and Juliano Foleiss. Automatic music genre classification in small and ethnic datasets. In *13th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, sep 2017.

[5] Mathieu Lagrange, Grégoire Lafay, Boris Defreville, and Jean-Julien Aucouturier. The bag-of-frames approach: a not so sufficient model for urban soundscapes. *Journal of the Acoustical Society of America*, 138(5):487–492, October 2015.

[1] https://github.com/vitorys/MusicGenreMetaClassifier

[6] Mikael Henaff, Kevin Jarrett, Koray Kavukcuoglu, and Yann LeCun. Unsupervised learning of sparse features for scalable audio classification. In *12th Proceedings of the International Conference on Music Information Retrieval*, 2011.

[7] Philippe Hamel, Simon Lemieux, Yoshua Bengio, and Douglas Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *12th Proceedings of the International Conference on Music Information Retrieval*, 2011.

[8] Jan Wülfing and Martin A Riedmiller. Unsupervised learning of local features for music classification. In *13th Proceedings of the International Conference on Music Information Retrieval*, 2012.

[9] Il-Young Jeong and Kyogu Lee. Learning temporal features using a deep neural network and its application to music genre classification. In *17th Proceedings of the International Conference on Music Information Retrieval*, 2016.

[10] Yandre M.G. Costa, Luiz S. Oliveira, and Carlos N. Silla. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied Soft Computing*, 52:28 – 38, 2017.

[11] S. Dubnov. Generalization of spectral flatness measure for non-gaussian linear processes. *IEEE Signal Processing Letters*, 11(8):698–701, Aug 2004.

[12] M. Hunt, M. Lennig, and P. Mermelstein. Experiments in syllable-based recognition of continuous speech. In *ICASSP '80. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 880–883, April 1980.

[13] Lawrence R Rabiner and Biing-Hwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[15] Bob L Sturm. A survey of evaluation in music genre recognition. In *International Workshop on Adaptive Multimedia Retrieval*, pages 29–66. Springer, 2012.

[16] Helge Homburg, Ingo Mierswa, Bülent Möller, Katharina Morik, and Michael Wurst. A benchmark dataset for audio classification and clustering. pages 528–531, 01 2005.

[17] Helge Homburg, Ingo Mierswa, Bülent Möller, Katharina Morik, and Michael Wurst. A benchmark dataset for audio classification and clustering. pages 528–531, 01 2005.

[18] Elias Pampalk, Arthur Flexer, and Gerhard Widmer. Improvements of audio-based music similarity and genre classificaton. In *Proceedings of the 6th International Conference on Music Information Retrieval*, 2005.

[19] Carlos Silla, Alessandro Koerich, and Celso Kaestner. The latin music database. pages 451–456, 01 2008.

A chord distance metric based on the Tonal Pitch Space and a key-finding method for chord annotation sequences

Lucas Marques[1]

[1]Computer Science Department – Federal University of Alfenas (UNIFAL-MG)
Av. Jovino Fernandes Sales, 2600 – 37133-840 Alfenas, MG

a14017@bcc.unifal-mg.edu.br

Abstract. *Music Information Retrieval (MIR) is a growing field of research concerned about recovering and generating useful information about music in general. One classic problem of MIR is key-finding, which could be described as the activity of finding the most stable tone and mode of a determined musical piece or a fragment of it. This problem, however, is usually modeled for audio as an input, sometimes MIDI, but little attention seems to be given to approaches considering musical notations and music-theory. This paper will present a method of key-finding that has chord annotations as its only input. A new metric is proposed for calculating distances between tonal pitch spaces and chords, which will be later used to create a key-finding method for chord annotations sequences. We achieve a success rate from* 77.85% *up to* 88.75% *for the whole database, depending on whether or not and how some parameters of approximation are configured. We argue that musical-theoretical approaches independent of audio could still bring progress to the MIR area and definitely could be used as complementary techniques.*

1 Introduction

In western music, tonality is a basic concept thoroughly studied by authors as Riemann [1] and Schoenberg [2], and could be defined in many ways. For example, a brilliant summarization of Schoenberg's thoughts on tonality and tonal function can be found on Carpenter [3]:

> Tonality for Schoenberg is not merely a certain collection of pitches of a scale, but more importantly, a kind of centricity. All pitches of a key-collection are related to a single tonal center, each in a specific way. The function of a single tone is signified by the degree of the scale it represents. The function of a chord depends upon its root, which is, in turn, the scalar degree upon which the chord is constructed. Tonality, then, is a set of functions of scalar degrees.

Since tonality is a fundamental concept of musical theory and many other information can be derived from the tonality of a piece, it is intuitive that within the Music Information Retrieval area, finding the tonality – also called key – of pieces would also be a fundamental problem. This paper presents a method of key-finding based on previous musical-theoretical work done by Lerdahl [4], a model of tonality named Tonal Pitch Space (TPS) which was corroborated by psychological experiments and matches musical intuition.

In Section 2 we discuss other methods of key-finding and theoretical models of tonality, and explain why we chose Lerdahl's Tonal Pitch Space. In Section 3, we focus on the TPS model, explain how it works and show its psychological and musical-theoretical background. Section 4 we introduces a new metric based on Lerdahl's model to calculate the distance between a tonal pitch space and any chord. Section 5 describes a method of key-finding utilizing that previous metric and Section 6 discuss its results on the test database created for this paper. More discussion will be brought about how audio-independent approaches could contribute to the MIR area in general in Section 7.

2 Related Work

One important problem in MIR is key-finding. Several models, algorithms and techniques were presented in the past considering the simple task of finding the global key of a determined piece of music or a local key within a subset of said piece – and, of course, methods of key-finding sensitive to context, with the objective of detecting changes of tonality (modulations). Chew [5] proposed, in 2002, a geometrical model of tonality that used a "Spiral Array" as a way to represent keys, chords, intervals and pitches. Pauws [6] describes a way of extracting the key of an audio source using chromagram computation. İzmirli [7] presented a model that uses a low-dimensional tonal representation. Hu [8] developed a probabilistic model to determine the key and also modulations on a MIDI database containing classical pieces from artists such as Bach, Mozart and Rachmaninoff. However, for this paper, we give a special attention to another model of tonality, proposed by Lerdahl [4]. Lerdahl's *Tonal Pitch Space* (TPS) is a model that correlates with empirical data provided by Krumhansl [9] and matches music-theoretical intuitions about tonality.

This paper is based upon the TPS model specially because we do not use audio from musical pieces as an input, but their chords annotations only. Thus, any model that uses audio as its main source of data would not serve our purpose. Since chord notation does not consider different octaves, solutions related to MIDI with notes from all octaves would also not bring any benefit to our objective. So, on top of all previous reasons to use Lerdahl's TPS, its simplicity is the main reason why we choose it. It completely matches the simplicity of chord annotations.

3 Tonal Pitch Space

The TPS is a model of tonality supported by empirical data from psychology [9] and matching human intuitions. It can be used to calculate the distance between all imaginable chords and it is based upon a notion of hierarchy between musical intervals in western music. That hierarchy is defined according to stability and what precisely fits with the experimental data provided by Krumhansl [9].

The best way to understand the TPS is looking at its structure. In Figure 1 we built the TPS of C major. The space is defined by five levels of stability, from the most to the less stable. The first one, *level a*, is the root level, which contains only the root of the TPS basic chord. In this case, since we are looking at the space of C major, it is C. This is in line with the fact that the most stable and consonant intervals are the octave and the unison.

Tonal Pitch Space

level a:	**C**											
level b:	**C**							**G**				
level c:	**C**				**E**			**G**				
level d:	**C**		**D**		**E**	**F**		**G**		**A**		**B**
level e:	**C**	**Db**	**D**	**Eb**	**E**	**F**	**Gb**	**G**	**Ab**	**A**	**Bb**	**B**

Figure 1: The TPS of C Major

Level b contains the root and the fifth interval which, in this case, is G. The second most stable interval reflects on the second level of the TPS. On the third level, *level c*, we have the triadic level, containing all the notes of the chord that generates the harmonic field represented by the TPS. The C major chord is composed of {C, E, G}. This set notation {} will be used ahead in this paper, and it is also very common to use the integers notation for the TPS (Figure 2). Next, *level d* is the diatonic level, containing the natural scale of the TPS. Here, we have the major scale of C. The last one, *level e*, is the most unstable of all, containing all 12 notes used in western music. We call it the *cromatic level*.

level a:	0											
level b:	0							7				
level c:	0				4			7				
level d:	0		2		4	5		7		9		11
level e:	0	1	2	3	4	5	6	7	8	9	10	11

Figure 2: The TPS of C Major with the numerical notation

4 Proposed Metric

In this section we discuss the metric proposed by this paper and its main differences between the metric proposed by Lerdahl himself when he presented the framework of the Tonal Pitch Space. Our metric differs specially within the two compared objects.

When introducing the concept of the TPS, Lerdahl [4] defined a method for calculating the distance between any two chords in the context of a key, and it is composed of two elements: circle-of-fifths distance and uncommon tones[1]. Lerdahl proposes a distance formula $d(x, y) = j + k$ where $d(x, y)$ is the distance between chords x and y, j is the minimal applications of the circle-of-fifths rule needed to transform x into y and k is the number of non-common pitch classes in the levels (a-d) within the basic space of y compared to the levels (a-d) in the basic space of x.

Our metric follows the same universal circle-of-fifths rule, but a different approach when considering common tones. Our formula is also conceived with the notion that there are 24 tonal pitch spaces (considering all 12 notes and major/minor qualities) and a much larger number of chords. Those chords are not necessarily part of a space, they don't necessarily fit into major/minor scales and they don't belong *a priori* to any harmonic field – which we imply that here are represented by the tonal pitch spaces. Thus, we do not compare two equal objects from a same metrical space. That said, our metric could be described as it follows:

$$d(S, x) = j + k$$

Where $d(S, x)$ is the distance between any tonal pitch space S and any chord x, j is the same previous circle-of-fifths rule[2] and k is the sum of the number of uncommon tones between the levels (a-c) of S and x and the difference between all notes of x and the *level d* of S. Figure 3 shows an example of the rule described previously. There we compare the distance between the TPS of C major and the chord G7. Check Figure 1 in need of remembering the levels of the TPS of C major.

Levels of *S*	**Levels of *x***
a: {C}	a: {G}
b: {C, G}	b: {G, D}
c: {C, E, G}	c: {G, B, D, F}
d: {C, D, E, F, G, A, B}	

Distance Calculation

uncommon(a) = 2, {C, G}
uncommon(b) = 2, {C, D}
uncommon(c) = 5, {C, E, B, D, F}

difference between *c* of *x* and *d* of S = 0, {}
circle-of-fifths(C → G) = 1

Total distance = 10.

Figure 3: Distance between the TPS of C major and the chord G7

[1] An uncommon tone is a tone present only in one of the two compared objects

[2] If a chord is non-diatonic, the circle-of-fifths rule returns a maximum value of 3. Major and minor qualities are also considered and, for example, the distance between C and Am on the circle-of-fifths is here considered as 1 instead of 0, since we have one step down to change from the major to the minor circle.

It is interesting to notice that this metric has a maximum value of 23, which can be obtained when comparing a C major tonal pitch space with a B chord containing all of the 12 notes of the cromatic scale. Such chord containing all notes is not musically practical, but this information could be useful for design, computation and application purposes of this metric. The minimum possible value is, as expected for a metric, a value of 0 when comparing a tonal pitch space to its basic chord, such as comparing that same previous C major tonal pitch space to its C major basic chord.

In the test database discussed in Section 5, we have found, comparing to the TPS of C major, chords on all distances from 0 to 21 - surprisingly close to the 23 maximum value. From 0 to 6 we have only chords with C as the root of the chord. The two chords that result in a distance of 21 are G#m7(9) and B6(9). The results of distances from chords to tonal pitch spaces match with musical-theoretical intuitions and reinforce the value of the tonal pitch space as a model for tonality. The results also match intuitions about chords within a given harmonic field.

Table 1 shows the distances between all seven degrees of C major. It is interesting to note how IV, V and VI have the same value and VII is closer than II is. This could be interpreted as the presence of VII being an indication of the tonality since it is present in only one harmonic field and it is also a chord of tension that suggests resolution on the tone center, matching musical intuitions.

5 A Key-Finding Method

Considering we have a metric to determine the distance between any chord and any tonal pitch space, the next step is to use it to develop a method to estimate tonality. We have done it following the premise that if a piece has a global key, the sum of the distances between the chords present in that piece should be minimum when comparing to the TPS of that global key. This was also assumed by de Haas [10] in his works based on Lerdahl's Tonal Pitch Space.

Thus, for each musical piece analyzed, we create 24 tonal pitch spaces (one for each major and minor of the twelve notes of the cromatic scale) and 24 variables of distance that are the sum of the distances between each chord present in the musical piece and each one of the tonal pitch spaces. We also have a multiplication factor MF for when the first or the last chord of the song is the same as the basic chord from the analyzed TPS – for example, if we are calculating the distances considering a TPS of Am and a song that starts or ends with Am.

Chord	Degree	Distance
C	I	0
Dm	II	14
Em	III	10
F	IV	9
G	V	9
Am	VI	9
Bdim	VII	13

Table 1: Distances from the chords of the C major harmonic field to a C major TPS

This multiplication factor improves the estimation results because musical pieces commonly start with the tonality to introduce it or end with it to resolve. From the set of 240 songs used for this paper, 180 songs start with the chord of the tonality, 145 songs end with the chord of the tonality and, within these two subsets, there is an intersection of 105 songs that start and also end with the chord of the tonality. Only 20 songs from the 240 do not start nor end with the chord that defines their tonality – this indicates how approximation features based on this information could benefit estimation of tonality. That said, the following formula describes the calculation of the distance between a musical piece and a tonal pitch space:

$$total\_dist = MF \sum_{i=1}^{n} TPS\_distance(S, c_i)$$

Where MF = OC, if only one chord – the first or the last – is the same as the basic chord of S; or MF = BC, if both chords are the same. The $TPS\_distance()$ function is the same we introduced at Section 4, applied to a tonal pitch space S and each chord c_i from the n chords of the piece. For the database used in this paper, we have tested values from 0.80 to 0.99 for OC and from 0.75 to 0.99 to BC – two digits precision. We have found that, from all permutations, the best results are obtained with values of 0.90 and 0.83 for OC and BC, respectively. For every configuration tested within these two ranges, there was an increase in the success rate of the prediction – indicating that perhaps this approximation step is almost never prejudicial to this method and should always be used.

6 Tests and Results

The objective of this paper was, *a priori*, the harmonic analysis of song chords available on websites such as Ultimate Guitar Tabs [11] and Cifra Club [12]. We chose the latter because of its vast repertoire, specially on brazilian music, such as Bossa Nova and Samba. We gathered[3], then, 240 songs of 104 artists from 25 different genres – Table 2 shows the distribution of songs per genre. Cifra Club pages provide the tonality of each piece. However, when analyzed thoroughly, that information was usually wrong.

Because of this lack of trust in the available information, we were forced to listen, play and search for information about each one of the 240 songs and their primary or global tonality. This was done in order to increase the reliability of the test database and guarantee ground-truth. This also helped deciding the first primitive values of

[3]For this part we simply downloaded the webpages of the songs and then parsed them to extract, from the HTML code, the chords and other relevant information as genre, title, etc.

Genres	Number of Songs
Alternative	17
Blues	1
Bossa Nova	11
Country	12
Dance	1
Disco	9
Folk	15
Funk	2
Grunge	2
Hard Rock	2
Heavy Metal	1
Indie	3
Jazz	6
Jovem Guarda	11
MPB	64
Pagode	4
Pop and Pop Rock	11
R&B	3
Progressive Rock	9
Rock and Rockabilly	31
Romantic	1
Samba	11
Soul	12

Table 2: Database's distribution of musical genres. Those genres were not checked for each song, but still indicate a good level of variety in the sample

the multiplication factor MF previously described – this process was basically noting how distant was the correct verified tonality from the one mistakenly estimated by our algorithm and whether the song had the first and/or last chord matching the tonality. This way, we could deduce how much should be the reduction applied by each one of the two parameters.

For the whole set of 240 songs, we show here two tests: one where the values of BC and OC were 1, i.e, without having the multiplication factor reducing the distance depending on the first and/or last chords. Without this feature, our method was able to achieve a success rate on estimating the correct tonality of 78.75%, with a 51 mistakes. This is already a promising success rate, and yet using different values – 0.82 and 0.90, tuned by hand – on a second test we are able to achieve 88.75% of success, with only 27 mistakes.

We also created categories of complexity[4] based on the number of different chords used on each song. This database is reasonably diverse on that matter, with songs going from having only three chords up to the most complex song – O Caderno, by Toquinho – having a total amount of 40 different parsed chords. With that in mind, we have created five arbitrary categories of complexity to test if the algorithm performs better or worse when estimating the tonality of songs with a lot or just a few chords.

[4]The categories of complexity were created to separate songs that most probably use chords not present in the harmonic field of the song.

Complexity Category	No. of Songs	Success Rate (%)
20+ chords	16	68.75
15–20 chords	44	70.45
9–14 chords	75	84.00
5–8 chords	67	79.10
less than 5 chords	54	77.77

Table 3: Categories of complexity, number of songs in each of them and success rates ($BC = OC = 1$).

Complexity Category	No. of Songs	Success Rate (%)
20+ chords	16	81.25
15–20 chords	44	88.63
9–14 chords	75	89.33
5–8 chords	67	88.06
less than 5 chords	54	88.89

Table 4: Success rates for each category when BC and OC are equal to 0.83 and 0.90, respectively.

The tests results are exposed in Table 3, without using the multiplication factors and also in Table 4 with the optimized configuration. Despite a considerable smaller success rate on songs with 20 or more chords, the method achieves promising results in all categories, specially when using the approximation feature of the multiplication factors.

7 Discussion and Conclusions

In this paper we have introduced a metric based on Lerdahl's Tonal Pitch Space to create a key-finding method. This method finds a single global tonality, but could possibly be used in local key-finding scenarios, given a subset of a piece. There are limitations from the simplicity of the input, given that they were only chord annotations without information of the duration of each one of them. It could also be discussed in further research a way of cross-validating the values of the parameters OC and BC in order to increase the generalization capacity of this small yet very relevant predictive part of the model.

However, even with simple chord annotations as input and every context based limitation, this method still managed to have great success rates when estimating the tonality of western popular music. This method could be used along with others, specially the ones with audio as input, to help in the task of finding the central key of a determined piece or.

It is also our argument that Music Information Retrieval could possibly benefit even more from theory-based and annotation-based methods. While audio and MIDI based computational and mathematical approaches seem to be dominant in this area, we might still have large space for progress for audio-independent techniques and approaches more rooted in musical-theoretical concepts.

References

[1] RIEMANN, H. *Harmony Simplified: or, The Theory of the Tonal Fuctions of Chords.* London: Augener Ltd., 1893.

[2] Schoenberg, A. (2010). *Theory of Harmony*. University of California.

[3] Carpenter, P. (1983). *"Grundgestalt" as Tonal Function.* Music Theory Spectrum, 5, 15-38. doi:10.2307/746093. p. 16-7.

[4] Lerdahl F. (1982) *Tonal Pitch Space*. In Music Perception: An Interdisciplinary Journal, Vol. 5 No. 3, Spring, 1988; (pp. 315-349) DOI: 10.2307/40285402.

[5] Chew E. (2002) *The Spiral Array: An Algorithm for Determining Key Boundaries.* In Anagnostopoulou C., Ferrand M., Smaill A. (eds) Music and Artificial Intelligence. ICMAI 2002. Lecture Notes in Computer Science, vol 2445. Springer, Berlin, Heidelberg.

[6] Pauws S. (2004) *Musical key extraction from audio*. In Proceedings of the 5th ISMIR, Barcelona, Spain., pages 96–99, 2004.

[7] Izmirli O. (2006) *Audio key finding using low-dimensional spaces.* In Proc. of the International Symposium on Music Information Retrieval (ISMIR), pages 127–132, Victoria, Canada, 2006.

[8] Hu, D., & Saul, L. K. (2009) *A Probabilistic Topic Model for Unsupervised Learning of Musical Key-Profiles.* In ISMIR (pp. 441-446).

[9] Krumhansl, C. L., & Kessler, E. J. (1982). *Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys.* In Psychological Review, 89(4), 334-368.

[10] de Haas W.B., Robine M., Hanna P., Veltkamp R.C., Wiering F. (2011). *Comparing Approaches to the Similarity of Musical Chord Sequences.* In: Ystad S., Aramaki M., Kronland-Martinet R., Jensen K. (eds) Exploring Music Contents. CMMR 2010. Lecture Notes in Computer Science, vol 6684. Springer, Berlin, Heidelberg.

[11] ULTIMATE GUITAR TABS - 1,100,000 songs catalog with free Chords, Guitar Tabs, Bass Tabs, Ukulele Chords and Guitar Pro Tabs!. [online] Available at: `https://ultimate-guitar.com` [Accessed 12 Jun. 2019].

[12] Cifra Club. (1996). Cifra Club - seu site de cifras e tablaturas. [online] Available at: `https://www.cifraclub.com.br` [Accessed 10 Jun. 2019].

Predicting Music Popularity on Streaming Platforms

Carlos V. S. Araujo[1]*, **Marco A. P. Cristo**[1] , **Rafael Giusti**[1]

[1]Federal University of Amazonas
Manaus, Brazil

{vicente, marco.cristo, rgiusti}@icomp.ufam.edu.br

Abstract. *Online streaming platforms have become one of the most important forms of music consumption. Most streaming platforms provide tools to assess the popularity of a song in the forms of scores and rankings. In this paper, we address two issues related to song popularity. First, we predict whether an already popular song may attract higher-than-average public interest and become "viral". Second, we predict whether sudden spikes in public interest will translate into long-term popularity growth. We base our findings in data from the streaming platform Spotify and consider appearances in its "Most-Popular" list as indicative of popularity, and appearances in its "Virals" list as indicative of interest growth. We approach the problem as a classification task and employ a Support Vector Machine model built on popularity information to predict interest, and vice versa. We also verify if acoustic information can provide useful features for both tasks. Our results show that the popularity information alone is sufficient to predict future interest growth, achieving a F1-score above 90% at predicting whether a song will be featured in the "Virals" list after being observed in the "Most-Popular".*

1 Introduction

The global entertainment market (movies, games, music, and television) is a billion-dollar industry. According to the Recording Industry Association of America – RIAA –, in 2018 the music industry was worth US$ 9.8 billion in the United States alone, 75% of which were due to streaming services and 11% to downloadable media[1]. Also according to the International Federation of the Phonographic Industry – IFPI –, in the same year the global industry was worth US$ 17.3 billion, 38% of which were due to streaming services and 16% to downloadable media[2].

It is no surprise that such market is fiercely competitive. Wikipedia lists over 1,400 record labels in the United States alone[3]. In face of such competition, the understanding of what makes an album or song successful is key information. It could be used to plan better marketing campaigns, to decide the best moment for the release of a new album or song, and to align the artists' effort with public interest in, e.g., genre, theme, etc.

The success of an album or single song may be assessed in several manners. The most common are probably by means of rankings, such as those provided by the American magazine Billboard, who has been evaluating the music market since the 1940s. Some of the most famous rankings from this magazine are the Hot 100 Songs and the Billboard 200[4]. The Billboard Hot 100 Songs list ranks the most popular songs from each week, while the latter ranks the most popular albums. Billboard also provides rankings that are specific to genres, countries, and distribution methods, as well as year-end rankings, which are versions of the previously mentioned charts, but for the popularity of music and albums over the entire year [1]. In spite of being fairly known to the general public, and also having been used for prediction of popularity in the past [2], Billboard charts are specific to the American market, and we do not employ them as data source in our work.

To assess the global market, we must focus on platforms that provide worldwide service. According to the IFPI, music streaming has become the most popular method of music distribution[2], finally surpassing physical media in 2017. Streaming services also provide several types of statistics concerning artists, albums, and scores, which we may explore to measure worldwide popularity. We shall focus our attention on the streaming platform Spotify, but we do note that our results may be extended to incorporate data from other streaming platforms, such as Tidal and Apple Music.

The problem of predicting success in the musical market has been addressed in the literature before, with Machine Learning algorithms providing the best results in general. Some authors employed acoustic features to create predictive models [3, 4], while others resorted to social network data [5, 6]. Our proposal relies only on historical data of popularity, which we measure with the aid of streaming services, to predict the continued success or popularity growth of a song. The major reason for seeking a model that is free from acoustic data is that such data may not always be available—Spotify, for instance, only provides a 30-second sample that may not be the most representative segment of a song. While social network data may be very rich, its collection and preprocessing may be very labour-intensive and expensive.

The remainder of this paper is as follows. In section 2 we discuss related work, mostly focusing on previous works that predicts popularity from music charts and social network data. Spotify's "Most-Popular" and "Virals" lists are presented in Section 3. The methodology used at this research is described in section 4, and our results are presented in Section 5. Finally, in Section 6 we draw our conclusions and discuss possibilities for future

*Supported by CAPES.

[1]www.riaa.com/wp-content/uploads/2019/02/RIAA-2018-Year-End-Music-Industry-Revenue-Report.pdf

[2]www.ifpi.org/downloads/GMR2018.pdf

[3]en.wikipedia.org/wiki/Category:American_record_labels. Visited 2019-06-10.

[4]https://www.billboard.com/charts

work.

2 Related Work

We have observed few popular approaches to the problem of music success prediction, despite it having received growing attention for several years. We remark two general strategies: the first uses social network data to assess current public perception and extrapolate how successful a song or album will be in the future. The second relies on acoustic information of previously successful songs to predict the success of a song or album.

The work by Kim et al. [7] is an example of the first strategy. The authors collected their data from the social network Twitter. Specifically, they analyzed posts (tweets) associated with the tags *nowplaying*, its shortened version *np*, and *itunes* (Apple's digital music sale platform). The task at hand was to predict whether a music would be successful, which the authors defined as being featured among the 10 first entries in the Billboard Hot-100 ranking. Despite having found a correlation of only 0.41 between the number of tweets and the number of weeks a song stayed in the ranking, the authors observed that a Random Forest classifier was able to achieve an accuracy of 0.90.

Araujo et al. [8] also made use of Twitter data. The authors collected tweets with mentions to 15 popular albums from 2016 and 2017. Their goal was to predict the number of Billboard units achieved by the albums and also their popularity according to Spotify. A Billboard unit can be reached with a single physical or digital sale or with $1{,}500$ music streams. Sentiment analysis was employed to verify whether tweet mentions were positive or negative. The authors observed linear correlation between positive tweets and Spotify popularity, but no correlation between negative tweets and Spotify popularity, neither between tweets of any polarity and Billboard data. The authors hypothesized that Billboard's choice of prioritizing physical sales does not reflect the modern market.

Among the works that resorted to acoustic data, Lee and Lee [3] observed 867 songs that made it to at least three consecutive weeks in Billboard's "Hot Rock Songs", a weekly ranking that features 50 entries of rock music. For each song, the authors collect a 30-second sample and extracted acoustic information, such as chroma, rhythm, and timbre. In addition to the acoustic data, they also employed information on whether the artist had been previously featured in the ranking. Those data were used to train a multi-layer perceptron classifier [9], and the authors' task was to predict how many weeks a song would remain in the ranking. Their model achieved an accuracy of 0.55 when only acoustic information was used and 0.54 when the model was trained with only information about previous appearances of the artist in the ranking. When both types of data were combined, the model achieved an accuracy of 0.59.

Karydis et al. [4] retrieved data associated with $9{,}193$ songs that were featured in at least one popularity ranking from the following sources between April 28th, 2013 and December 28th, 2014: Billboard, Last.fm, and Spotify. Additionally, they retrieved data from songs of the albums in which these popular tracks were released. This resulted in a data set of popularity scores and acoustic information of $23{,}385$ songs. Their goal was to employ knowledge of the most successful songs from past albums to predict which song will be the most successful from an unseen album. The authors employed two temporal-data models, namely a non-linear autoregressive network classifier (NAR) and its variation with exogenous inputs (NARX). The authors reported precision of 0.46 and accuracy of 0.52.

In addition to the previously discussed strategies, Arakelyan et al. [10] compiled data from SongKick[5] about live performances and festivals. Their task was to predict whether artists feature in those performances and festivals would sign contracts with major record labels. They employed logistic regressors and reported precision of 0.39. Steininger and Gatzemeier [11] analyzed data from live performances in Germany and their task was to predict whether songs from the observed artists would be featured among the 500 most popular German songs in 2011. Using Partial Least Squares Structural Equation Modeling (PLS-SEM), the authors reported an estimated precision of 0.43.

3 Spotify's Lists

We collected our data from the streaming platform Spotify. Spotify is the third largest music streaming platform, according to Forbes[6]. Spotify publishes daily lists of popular and "viral" songs which, according to Kevin Goldsmith[7], a company's former vice-president of engineering, are constructed in the following ways: the "Most-Popular" list ranks songs according to the total number of streams in the previous day, while the "Virals" list ranks song according to the growth in number of streams.

From Goldsmith's description we draw that a song will remain among the "Virals" if its number of listeners is constantly rising, which implies that the song is reaching a broader public than what is usual for a particular artist. It seems reasonable to assume that virality is an event desired by artists who want to expand their audience. However this definition of virality also means that already successful artists will find it more challenging to hit the "Virals" list than the "Most-Popular" list. On the other hand, less famous artists will tend to find it harder to reach the "Most-Popular" list, being more likely to be featured among the "Virals".

Our goal is to predict whether an already-popular artist may experience sudden growth in public interest. We

[5] `https://www.songkick.com/`

[6] `https://www.forbes.com/sites/hughmcintyre/2018/05/25/the-top-10-streaming-music-services-by-number-of-users`. Visited 2019-06-11.

[7] `https://www.quora.com/What-are-Spotifys-US-Viral-50-and-Global-Viral-50-charts-Are-these-just-daily-charts`. The answer is visible only to logged in users. Visited at 2019-06-11.

do so by making use of data from the “Most-Popular” list to predict appearances in the “Virals”. The converse may also be of great interest. Once a piece of work reaches the status of viral, is the sudden spike in popularity merely incidental and temporary? Or does the regular audience remains more expressive in the long term? To answer this, we predict whether a music that is featured in the “Virals” list will be featured in the “Most-Popular”.

4 Methodology

In this section we present our experimental method. We propose four models for predicting appearances in Spotify’s “Virals” list from appearances in its “Most-Popular” list, and vice versa. The first employs previous data from one list to make predictions in the other, and the second extends this model with acoustic information. The third model makes predictions using acoustic information only. And the fourth model is a baseline, which only counts appearances in one list and predicts appearances in the other if the song has surpassed a threshold in the first list.

4.1 The Classification Problem

Our experimental method is outlined in Figure 1. It consists of the following steps: data collection, extraction of acoustic features, baseline generation, data set preparation, models training and testing, and analysis of results.

Before we discuss how we collected the data and constructed the models, we shall define our classification problem, which we divide in two phases.

During the *assessment phase*, we collected data for days $D_1, D_2, \ldots, D_n$. We shall discuss the data in Section 4.2 but, for the moment, it suffices to say that the data for each day D_i is represented as a pair of lists (P_i, V_i) that contains information for the 50 “Most-Popular” songs and the 50 “Virals” songs of that day. The parameter n is the number of days for which we collected training data.

During the *prediction phase*, we aim to answer the following questions:

1. For every song featured in the “Most-Popular” list $P_{t>n+k}$, will it also be featured in the “Virals” list V_{t+1}?
2. For every song featured in the “Virals” list $V_{t>n+k}$, will it also be featured in the “Most-Popular” list P_{t+1}?

In both cases, t and $t+1$ are the days for which we want to make predictions. The inequality $t > n+k$ simply means that the assessment phase and the prediction phase do not have overlapping days, and that the prediction phase starts k days after the assessment phase has ended. This gap is imposed to avoid overlapping information from the lagged features, discussed in section 4.4, therefore k is a hyperparameter in our models.

The models themselves are discussed in depth in Section 4.5. At this time, we want to make the reader aware that the models were trained with data from the assessment phase only. This restriction allows us to predict several days after the assessment phase, allowing for early prediction of popularity and “viralization” phenomena.

4.2 Data Collection

Data from the “Virals” and “Most-Popular” lists were collected using Spotify’s Web API[8]. The data were collected on a daily basis between November, 2018 and January, 2019.

For each daily list, we collected information for nine fields made available by the API. Namely, the rank and the date of the ranking, the names of the artists and of the song, date of release of the song, duration in milliseconds and a URL for a 30-second sample of the song. We note that the URL was not available for roughly 3% of the songs feature in our data, so we removed any rows where this field was empty. Additionally, each song has an “explicit” flag, which indicates whether it contains profanity, and a popularity score, which is a value in the $[0, 100]$ interval that reflects how popular the song is.

Finally, we downloaded the 30-second sample of each song and extracted five acoustic features. The following features were extracted with the Python package LibROSA [12]:

1. Mel-Frequency Cepstral Coefficients (MFCC): obtained from the cepstrum of the compressed mel representation of the signal. The MFCC is probably one of the most often used features in speech processing, and is an expressive low-dimensional representation of a signal. In this work we used 13 coefficients per song;
2. Spectral Centroid: the centroid of each frame of a magnitude spectrogram that has been normalized and treated as a distribution over frequency bins;
3. Spectral Flatness: a measure of noise-like a sound is, as opposed to being tone-like [13];
4. Zero Crossings: the number of times a waveform changes sign;
5. Tempo: number of beats per minute.

4.3 The Baseline

The baseline is a low-effort approach to answer questions 1 and 2 using the least amount of available information. It should be straightforward and, therefore, easily surpassed by a model specifically designed to make popularity predictions. The process of constructing the baseline is closely related to how we make the data set, so we shall present it before the actual models.

During the assessment phase, to each individual song is assigned a “Popularity-Presence” score and a “Viral-Presence” score, which is the number of times that a particular song was featured in a list. For instance, if song x_j was featured in the “Most-Popular” list for two consecutive weeks in November, and then for 5 non-consecutive days in December, then its “Popularity-Presence” score

[8] https://developer.spotify.com/documentation/web-api/

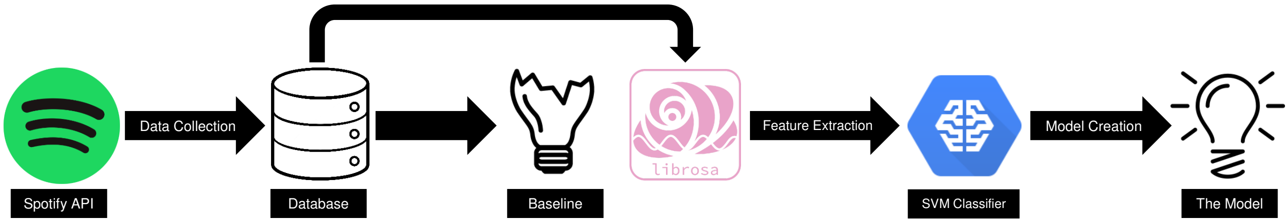


Figure 1: Outline of our experimental method.

will be $p_j = 19$. Similarly, if the same song appears in 13 daily "Viral" lists, then its "Viral-Presence" score will be $p_v = 13$.

These scores are used to define two thresholds that will be used by the baseline model. The "Most-Popular" threshold θ_p is defined as the the lowest value of "Popularity-Presence" score a song must have to be considered probable to be featured again in that list. We empirically set θ_p to be the median of all "Popularity-Presence" scores after looking at the distribution of the scores. Similarly, we defined the "Viral" threshold θ_v as median of the "Viral-Presence" scores.

As explained in Section 4.1, the classification problem we are tackling requires predicting whether a song will appear in one list after being observed in the other. Therefore, we define the popularity baseline as follows. During the prediction phase, for some $t > n$, if a song x_j appears in the "Most-Popular" list P_t and p_j is at least as large as the popularity-baseline threshold, then it will appear in the "Viral" list V_{t+1}. In other words, a song featured in today's "Most-Popular" will be featured in tomorrow's "Viral" if it was featured more than the popularity-baseline threshold during the assessment phase.

Similarly, the viral baseline will predict that a song x_j that appears in the "Viral" list V_t will be featured in the "Most-Popular" list P_{t+1} if v_j is at least as large as the viral-baseline threshold.

4.4 Data Set Creation

The data set was created from the lists collected during the whole period in a number of steps. The first step was the insertion of cross-information between the two types of lists. That is, for entries from the "Most-Popular" list we added relevant "Viral" information and vice versa. The cross-information for each list P_i and V_i varies daily, and includes the position of a song in the other list, the number of consecutive days the song has been featured in the other list. Because the classification task is to predict whether a song that appears in one list will also appear in the other list on the next day, we add to each "Virals" entry a flag indicating whether it is featured in the "Most-Popular" list on the next day and vice versa. Specifically, that flag is the class label. Furthermore, if a song has already been featured in the other list at least once prior to the i-th day, then we also add the date in which it first appeared.

With exception of the 30-second sample URL, no attributes have missing values when collected through the API. However, the insertion of cross-information may cause missing values to appear. This could happen, for instance, if a song from the "Virals" list has never appeared in the "Most-Popular" in the assessment phase. Because of the nature of those data, we simply replace missing data with zeros.

Finally, all non-numeric fields were transformed into categories through one-hot encoding [14]. And because the URLs to the 30-second samples were only required to extract acoustic features, they were removed from the data set at this point.

To capture temporal patterns present in the data, we add lagged features to each instance [15]. In particular, we lag the rank of each song in each list. For instance, at day D_3 in the "Most-Popular" list, an entry $s_{3,i}$ for some song x_j has the features $r_{3,i}$, $r_{2,i}$, and $r_{1,i}$, which express the position of x_j in the "Most-Popular" list during days D_3, D_2, and D_1. Similarly, we lag the rank of each song in the "Virals" list.

4.5 Data Prediction

Before instantiating any models, we first partitioned the data into training and test sets. The data of the lists collected between November 1st and December 31st of 2018 were used to make the training set. There were several test sets, which reflects from our experimental setting of evaluating the models for short-term prediction, mid-term prediction, and long-term prediction. The data used to build the test sets were collected between January 3rd and January 30th of 2019. The two first days of January were not included neither in the training nor in the test because lagged features cause information overlap with data from the training set.

For each list, we produced four test sets. The first set, which we shall refer to as 1st Week, contains data from the lists collected between January 3rd through January 9th. The second set, namely 2nd Week, contains data from the lists collected between January 10th through January 16th. Similarly, the next two sets, 3rd Week and 4th Week, contain data collected until January 23rd and 30th, respectively. Each test set has approximately 350 entries for each list—fewer than that if any entry was removed due to not containing a URL to its 30-second sample.

Recall that, for each day D_t, the class is whether the song will be featured in the other list on day D_{t+1}. Therefore the instances associated with the first day of the 1st Week in the "Virals" prediction, for example, are songs from the “Most-Popular” list from Jan. 3. Notice that we can only make “Virals” claims for songs that appear in the “Most-Popular” list the previous day. Therefore, a true positive will be a song from the “Most-Popular” that we claimed to appear in the “Virals” list the next day and indeed appeared in the “Virals” list the next day, while a true negative will be a song that appears in the “Most-Popular” in one day but we claimed to *not* appear in the “Virals” the next day—and verify to have predicted correctly. Similarly for “Most-Popular” predictions we can only make claims for songs that were featured in the “Virals” list the day before.

To make this prediction, we trained a Support Vector Machine (SVM) classifier [16]. The SVM is an instance-based classifier that projects the training samples into a space of higher dimensionality, where it assumes to have a representative layout of the original space. In this projection, the SVM attempts to find the hyperplane that best separates the classes, effectively dividing the decision space into two subspaces. When classifying a new sample, the SVM projects its features into the same high-dimensional space and verifies on which subspace the projected instance “falls”, and then assigns it the class label associated with that subspace [17].

The model was trained and tested with the library “scikit-learn” [18], which implements its own version of the SVM inducer based on the LibSVM implementation [19]. We used the *RBF* kernel and kept recommended default values for all parameters.

The results obtained were evaluated using seven distinct metrics. Namely, accuracy, Area Under the ROC Curve (AUC), Matthews Correlation Coefficient (MCC), sensitivity, F1 Score, precision and specificity.

5 Results

In total, we collected 92 instances of the “Most-Popular” and “Viral” lists from Spotify, resulting in 9,200 rows of data. We removed 132 rows from the “Virals” and 200 rows from the “Most-Popular” lists that were lacking the URL for the 30-second sample. The remaining 8,868 rows contain data associated with 400 unique songs, out of which 231 are featured only in the “Virals” list and 116 are featured in the “Most-Popular” lists. Only 53 songs appeared in both lists.

The confusion matrices for each test round are available at `http://bit.ly/sbcmpaper`. In Figure 2 we show the confusion matrix of our proposal for the fourth week of “Most-Popular” prediction. This matrix represents the round that gave the highest AUC score, with only six misclassified instances. It is also noticeable that the number of negative instances is nearly twice the number of positive instances. This is also observed in the “Virals” list, as shown in Figure 3. We note that this is the worst result achieved by the acoustic features-based model. This experiment also refers to the fourth week of the prediction phase.

The results for the baseline are given in Table 1

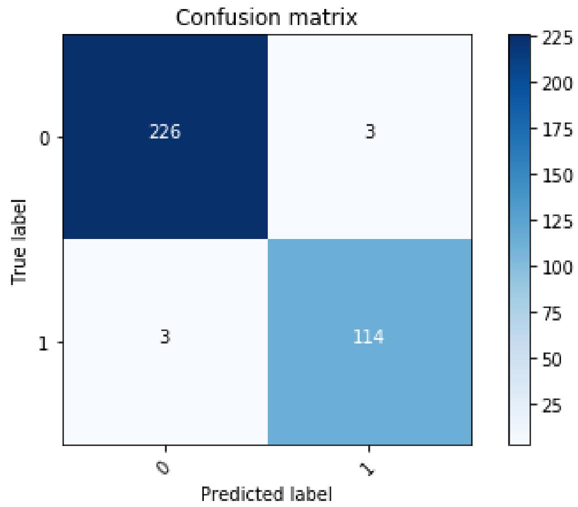


Figure 2: Confusion matrix of the proposed model with only previous list data for predicting the 4th week of the “Most-Popular” list.

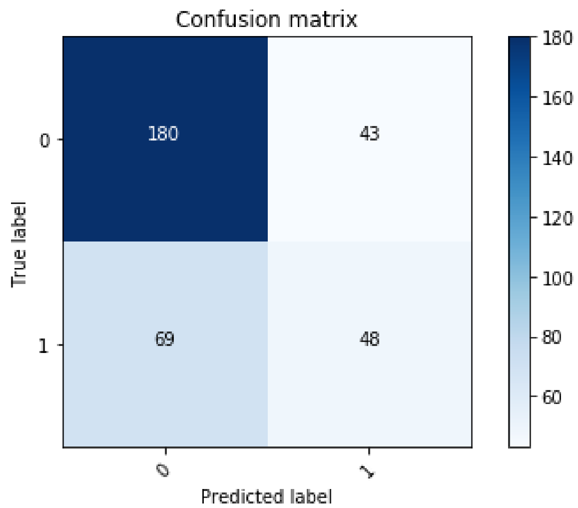


Figure 3: Confusion matrix of the acoustic feature-based model for predicting the 4th week of the “Virals” list.

and the results for the model that uses only the lists information are given in Table 2. The results for the model trained only with acoustic features are given in Table 3 and the results for the model that makes use of all types of data are given in Table 4.

Initially, we note that the two baselines showed very different behaviors. The “Virals” baseline has MCC values much closer to 0 than 1, which indicates low predictive power. Furthermore, the precision is below 0.5 in the fourth week. This is expected, of course, considering this is the baseline.

On the other hand, the baseline is surprisingly effective in predicting “Most-Popular” songs. The “Most-Popular” list is much less volatile than the “Virals”, which featured 68% more individual songs than the former for the same period. We however note that the performance of this

	Viral				Top			
	1st Week	**2nd Week**	**3rd Week**	**4th week**	**1st Week**	**2nd Week**	**3rd Week**	**4th week**
Accuracy	0.7062	0.7560	0.6932	0.6416	0.9475	0.9296	0.8675	0.8526
AUC	0.6970	0.7465	0.6691	0.6217	0.9182	0.8856	0.8070	0.7862
MCC	0.3715	0.4662	0.3293	0.2196	0.8746	0.8247	0.7148	0.6741
Sensitivity	0.6731	0.7228	0.5982	0.5214	0.8447	0.7835	0.6140	0.5812
F1 Score	0.5858	0.6404	0.5630	0.4959	0.9063	0.8636	0.7609	0.7273
Precision	0.5185	0.5748	0.5317	0.4729	0.9775	0.9620	1.0000	0.9714
Specificity	0.7210	0.7702	0.7401	0.7031	0.9917	0.9877	1.0000	0.9913

Table 1: Performance of the baseline models.

	Viral				Top			
	1st Week	**2nd Week**	**3rd Week**	**4th week**	**1st Week**	**2nd Week**	**3rd Week**	**4th week**
Accuracy	0.9436	0.9375	0.9410	0.9353	0.9650	0.9824	0.9849	0.9827
AUC	0.9113	0.9073	0.9107	0.9161	0.9473	0.9784	0.9781	0.9806
MCC	0.8683	0.8496	0.8689	0.8558	0.9163	0.9568	0.9668	0.9613
Sensitivity	0.8269	0.8317	0.8214	0.8547	0.9029	0.9691	0.9561	0.9744
F1 Score	0.9005	0.8889	0.9020	0.9009	0.9394	0.9691	0.9776	0.9744
Precision	0.9885	0.9545	1.0000	0.9524	0.9789	0.9691	1.0000	0.9744
Specificity	0.9957	0.9830	1.0000	0.9776	0.9917	0.9877	1.0000	0.9869

Table 2: Performance of the models that do not use acoustic features.

	Viral				Top			
	1st Week	**2nd Week**	**3rd Week**	**4th week**	**1st Week**	**2nd Week**	**3rd Week**	**4th week**
Accuracy	0.8101	0.8185	0.7552	0.6706	0.8630	0.8798	0.8253	0.8121
AUC	0.7695	0.7714	0.6996	0.6087	0.7718	0.7886	0.7456	0.7264
MCC	0.5482	0.5582	0.4237	0.2333	0.6743	0.7030	0.6299	0.5816
Sensitivity	0.6635	0.6535	0.5357	0.4103	0.5437	0.5773	0.4912	0.4615
F1 Score	0.6832	0.6839	0.5911	0.4615	0.7044	0.7320	0.6588	0.6243
Precision	0.7041	0.7174	0.6593	0.5275	1.0000	1.0000	1.0000	0.9643
Specificity	0.8755	0.8894	0.8634	0.8072	1.0000	1.0000	1.0000	0.9913

Table 3: Performance of the models that use acoustic features only.

	Viral				Top			
	1st Week	**2nd Week**	**3rd Week**	**4th week**	**1st Week**	**2nd Week**	**3rd Week**	**4th week**
Accuracy	0.9080	0.8690	0.8584	0.7971	0.9417	0.9179	0.8554	0.8295
AUC	0.8510	0.7822	0.7857	0.7132	0.9057	0.8619	0.7895	0.7541
MCC	0.7871	0.6895	0.6868	0.5463	0.8614	0.7961	0.6888	0.6193
Sensitivity	0.7019	0.5644	0.5714	0.4444	0.8155	0.7320	0.5789	0.5214
F1 Score	0.8249	0.7215	0.7273	0.6012	0.8936	0.8353	0.7333	0.6740
Precision	1.0000	1.0000	1.0000	0.9286	0.9882	0.9726	1.0000	0.9531
Specificity	1.0000	1.0000	1.0000	0.9821	0.9958	0.9918	1.0000	0.9869

Table 4: Performance of the models that use all available data.

model quickly degrades for the 3rd and 4th weeks.

When we analyze all results, we verify that our proposed model has the highest metrics when only the list information is used to make the classification, with the exception of specificity and precision. Compare those metrics in Table 2 and Table 4 for the 1st and 2nd weeks. This result suggests that the past performance of a song is a good indicator of whether it will experience a spike in popularity, and that using acoustic features may improve the performance of the classifier, but is not required.

We note that the model based on the lists past information shows little degradation as the time window shifts away from the assessment phase. Notice in Table 2 that there is little reduction in all metrics for prediction in both lists. In the worst case, the precision dropped 3.6% in the "Virals" list and the accuracy dropped 0.8% in the same list. On the other hand, it is fair to say that, in spite of being consistent for all weeks during the prediction phase, it does not seem to substantially surpass the baseline in the first two weeks, as its accuracy and AUC are very similar to the baseline in the aforementioned rounds.

We do note that the model based exclusively in acoustic information obtained the worst results. Its performance was worse than the baseline according to many metrics for predictions in the "Most-Popular" list. We also note that this model degrades rapidly as the time window moves away from the assessment phase, declining 32.49% in MCC for predictions in "Virals". At best, the acoustic model dropped 7.8% in specificity for the "Virals" list and 0.8% in specificity for the "Most-Popular" list. Recall that specificity is the fraction of negative instances correctly identified. This suggests that the acoustic model tends to be over-optimistic in predicting that a song will go viral. Accuracy drops by approximately 12% in both lists and precision drops by approximately 7% and 3% in "Virals" and "Most-Popular", respectively.

The incorporation of acoustic information to the popularity model has proved to give less exciting results than expected. While the performance was not as low as when only acoustic information was used, it did not surpass the model that relies only on data from past "Virals" and "Most-Popular" lists. This is a surprising result, which deserves further analysis. We do note, however, that this model did not degrade as much as when only acoustic information was used, which do suggest that this model could be modified to properly predict success several days after the assessment phase has finished.

6 Conclusion and Future Work

In this paper we have discussed an approach to predict whether a song will have a spike in popularity (i.e., if it will "go viral") or if a song will be consistently popular. We define that a song has experienced a spike in popularity if it appears in Spotify's "Virals" list, and that a song is consistently popular if it is featured in Spotify's "Most-Popular" list. Our model predicts whether a song that is featured in one list will be featured in the other the next day. This approach has been chosen because it is generally more difficult for famous artists to suddenly experience a spike in popularity and be featured in the "Virals" list, while less successful artists tend to find it harder to appear in the "Most-Popular" list. We expect that to make our model useful in both situations.

There are several works in the literature that deal with the problem of predicting song or album success. Our approach has the advantage of requiring only data from the song past popularity. While acoustic information may be used, it is not necessary, and in fact our experiments have shown that results are superior when only previous popularity knowledge is used to train the model.

Because we did not find a suitable baseline for our experimentation model, we also propose a baseline in our paper. The proposed baseline considers how often a song has been featured in the "Most-Popular" and "Virals" list with respect to other songs, and predicts that they will be successful or experience a spike in popularity if they are more frequent than the median of the frequencies in the appropriate list. Our model was observed to be consistent when the prediction window is further from the training date, and achieved high statistics in all metrics chosen for evaluation.

We conclude that the proposed model is successful in employing popularity information to predict if a song will "go viral", and to predict if "viralization" phenomena will be followed by consistent growth in popularity for a given song.

We do note that, while our work is limited by to the extent of the data provided by Spotify, we believe our proposed model may be extended to other platforms. For example, some platforms provide lists of "trending" songs, or artists, which we could consider equivalent to Spotify's "Virals" list. On the other hand, streaming platforms usually provide lists of popular artists, albums, or songs, therefore they could be directly used by our model.

Furthermore, while we do aim at exploring models that do not require social network information, we note that our model might benefit from them. And we intend to use that type of data to garner more information as means of more accurately establishing the popularity of a song or artist.

This paper is part of a series related to music success prediction. On previous works we demonstrate the importance of social networks for the success of an album [8] and how the process of artist collaboration in Rap Music works [20]. We also identified influence factors on the popularity of musical genres [21].

References

[1] Chris Molanphy. How the hot 100 became america's hit barometer, 2013.

[2] Cristian Cibils, Zachary Meza, and Greg Ramel. Predicting a song's path through the billboard hot 100', 2015.

[3] Junghyuk Lee and Jong-Seok Lee. Predicting music popularity patterns based on musical complexity and early stage popularity. In *Proceedings of the Third Edition Workshop on Speech, Language & Audio in Multimedia*, SLAM '15, pages 3–6, New York, NY, USA, 2015. ACM.

[4] Ioannis Karydis, Aggelos Gkiokas, Vassilis Katsouros, and Lazaros Iliadis. Musical track popularity mining dataset: Extension & experimentation. *Neurocomputing*, 280:76 – 85, 2018. Applications of Neural Modeling in the new era for data and IT.

[5] Vasant Dhar and Elaine A. Chang. Does chatter matter? the impact of user-generated content on music sales. *Journal of Interactive Marketing*, 23(4):300 – 307, 2009.

[6] Benjamin Shulman, Amit Sharma, and Dan Cosley. Predictability of popularity: Gaps between prediction and understanding. In *Tenth International AAAI Conference on Web and Social Media*, pages 348–357, 2016.

[7] Yekyung Kim, Bongwon Suh, and Kyogu Lee. #nowplaying the future billboard: Mining music listening behaviors of twitter users for hit song prediction. In *Proceedings of the First International Workshop on Social Media Retrieval and Analysis*, SoMeRA '14, pages 51–56, New York, NY, USA, 2014. ACM.

[8] Carlos V.S. Araujo, Rayol M. Neto, Fabiola G. Nakamura, and Eduardo F. Nakamura. Predicting music success based on users' comments on online social networks. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and*

the Web, WebMedia '17, pages 149–156, New York, NY, USA, 2017. ACM.

[9] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[10] Shushan Arakelyan, Fred Morstatter, Margaret Martin, Emilio Ferrara, and Aram Galstyan. Mining and forecasting career trajectories of music artists. In *Proceedings of the 29th on Hypertext and Social Media*, HT '18, pages 11–19, New York, NY, USA, 2018. ACM.

[11] Dennis M Steininger and Simon Gatzemeier. Using the wisdom of the crowd to predict popular music chart success. In *Proceedings of the 21st European Conference on Information Systems*, page 215, 2013.

[12] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

[13] S. Dubnov. Generalization of spectral flatness measure for non-gaussian linear processes. *IEEE Signal Processing Letters*, 11(8):698–701, Aug 2004.

[14] David Harris and Sarah Harris. *Digital design and computer architecture*. Morgan Kaufmann, 2010.

[15] Jason Brownlee. *Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future*. Jason Brownlee, 2017.

[16] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[17] R. Giusti, D. F. Silva, and G. E. A. P. A. Batista. Improved time series classification with representation diversity and svm. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1–6, Dec 2016.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[20] Carlos V.S. Araújo, Rayol M. Neto, Fabiola G. Nakamura, and Eduardo F. Nakamura. Using complex networks to assess collaboration in rap music: A study case of dj khaled. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*, WebMedia '17, pages 425–428, New York, NY, USA, 2017. ACM.

[21] Carlos V. S. Araujo and Eduardo F. Nakamura. Identification of most popular musical genres and their influence factors. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, WebMedia '18, pages 233–236, New York, NY, USA, 2018. ACM.

Studio and Laboratory Reports

NESCoM Research Report (2019)

Leandro Costalonga[1] , Daniel José Custódio Coura [2], Marcus Vinicius das Neves[3], Fabiano Araújo Costa[4], Helder Roberto de Oliveira Rocha[5].

NESCoM – Núcleo Espírito-santence de Computação Musical

Universidade Federal do Espírito Santo

[1, 2] Campus São Mateus
BR 101 Km 60 – 29932-540 São Mateus, ES

[3,4,5] Campus Goiabeiras
Av. Fernando Ferrari, 514 - Goiabeiras, Vitória - ES | CEP 29075-910

leandro.costalonga@ufes.br, daniel.coura@ufes.br, creed.mvmn@gmail.com, fabiano.costa@ufes.br, helder.rocha@ufes.br

Abstract

The NESCoM is a multidisciplinary research center formed by musicians, engineers and computer scientists. The main research interest lies with sonology, audiotacticle musical analysis, ubiquitous music, interactive multimedia installations, and the design of computer music technology in general. Overall, the common ground for the NESCoM projects lies with the human-aspects, both cognitive and motor, behind a musical activity. This can come, for instance, in the shape of an audiotactile analysis of musical interaction applied to a new digital musical interface designed to overcome human physical constraints or the composition of a cinema soundtrack based on perceptual models of the audience. In this paper, it is reported a short description of the ongoing projects of the NESCoM and the future works.

1. Introduction

In 2010, the Computer Music Committee (CECM) of the Brazilian Computing Society (SBC) approved Vitória-ES as the host city of the XIII Brazilian Symposium on Computer Music (SBCM). At that time, there was little organized research initiatives in the field happening at the local universities. Moved by the necessity to organize such event, an interinstitutional local committee was formed. The relationship between the host institutions grew stronger during the process of organizing such event, up to a point that some of researchers decided to collaborate more extensively soon after the event – the NESCoM was born. However, it was not until 2013 that it was officially registered as a formal research group of Federal University of Espírito Santo (UFES) at CNPq. The group is formed by musicians, electrical engineers, and computer scientists. The members are mainly spread between the Music Department (Campus Goiabeiras/Vitória-ES) and the Computing and Electronics Department (Campus São Mateus-ES). Despite the challenges of the remote collaboration, being geographically apart gives us the opportunity to stretch out to a vaster audience and to serve students (undergrads and postgrads) from different degrees, such as: music, computer science, computer engineering, and electrical engineering.

The main research lines are: soundtrack composition and sound-design for cinema and digital games, audiotactile musical analysis and applications, interactive multimedia installations, ubiquitous music, and new interfaces for musical expression, as detailed and exemplified by the ongoing projects.

2. GEXS – The Sound Experimentation Group

The Sound Experimentation Group (GEXS) emerged in 2011 with the creation of Atari Sound Perfomance (ASP), a partnership between Marcus Neves - founder of GEXS - and the sound artist Herbert Baioco, who was at that time a substitute professor of the Music Course of UFES.

The Atari Sound Perfomance (ASP) used circuit bending technique to perform real time audio and video processing using an old Atari video-game. The ASP was presented in Festivals in Rio de Janeiro, Paraná, São Paulo and Espírito Santo. In 2012, with the participation of three students of the course, was created the first sound installation, (en) roll, exhibited in May of that year at Galeria Universitat Universitario of UFES.

In 2013 the group restructured itself during the first semester and resumed activities from August of that year. Two lines of work were established: [1] Study and creation of individual and collective performances within the field of Experimental Music and Sonic Arts. In this line of work, collective performances, compositions and acoustic music concerts, and sound production for radio programs were developed. Among the performances, Brinks (2014) went beyond the limits of the academic sphere, being performed at the National Meeting of University Composers - ENCUn (2014), in São Paulo / SP and Fábrica.Lab (2014), in Vitória / ES. In addition to the collective practice, the development of individual musical compositions is also encouraged and later compiled into discs released annually. During the period between 2014 and 2019, several participations took place at the Electronic Language International Festival - FILE in the form of sound works and musical pieces composed by the students Joceles Bicalho, in 2015 and in 2016 in partnership Marcus Neves, and Giuseppe Cavatti, in 2019. The three pieces derived from this musical production performed within the GEXS. [2] Study of sound language for audiovisual works, especially for cinema and series; In this line we study [a] concepts, forms and techniques applied in the sectors of the production chain of the area of sound for audiovisual works: sound capture, soundtrack composition, sound effects production, sound editing, mixing and sound design; [b] film analysis of audiovisual language applied to software such as eAnalysis and iAnalysis, developed by the French researcher Pierre Couprie. From 2015 until the present day, the group has participated in several university and professional films in the audio sectors, especially in the post-production and in the composition of pre-established musical track or in the real-time performance for movie accompaniments, such as in the case of The Marriage of Clarice and Bataille (2017), by the artists Aline Dias and Júlia Amaral, who later would derive as a sound installation and radio-soap opera, and in the performance for the opening of the Sci-Fi Sesc Glória Cultural Center Festival (Vitória / ES). Some research works also derived from this axis either from the Scientific Initiation Program of UFES or in the form of papers for academic events in the area of sound and music for film and audiovisual.

This project is coordinated by Marcus Vinicius Marvila das Neves and, currently, the group consists of the following students: Yasmin Marques, Gabriel Madeira, Eduardo Buback, Gisele Bernardes, Jennifer Nogueira, Alessandra Felix, Giuseppe Cavatti, Gabriel Amorim, Dyone Cipriano, Mario Schiavini, Bruno Hanstenreiter. The professor Alexsandro Meirelles (UFES) also collaborates as an associate researcher.

3. Audiotactile Analysis

According to Vincenzo Caporaletti's [1], the general criterion for identifying audiotactile music is the existence of the mediological pair "audiotactile principle [ATP] + neo-auratic encoding [NAE]" in the formative process of art music. This simple yet dense formula reveals that we may find, in this music's poietic dimension, important aesthetic precepts associated with groove, swing, and the propulsive and depulsive formative energy etc. Such criteria are induced by the awareness of the technological inscription and phonographic fixation of these values, and also by acknowledging the relevance of this process as a constitutive modality of music as a work of art.

This project aims to develop studies that problematize conceptual and referential aspects about improvisation, groove, and musical interaction in Audiotactile Music (jazz, rock, pop, rap, Brazilian popular music, improvised music, world music, etc. Moreover, it aims to build a database on the micro and macro structural dimension of groove in Brazilian (urban / popular) songs that are re-assignable to the taxonomic model of the songs of audiotactile expression[1][2]. In other words, it seeks to collect relevant data for the investigation of musicological problems associated with the formative processes (production and reception) of a groovy, improvisational, extemporaneous and interactional nature in the music produced in Brazil since the beginning of the 20th century, where the phonographic record is the textual status of the musical work, considering the factors intrinsic to the inter and transcultural dynamics characteristic of such processes. Specifically, we intend to analyze a corpus of phonograms produced in Brazil since the beginning of the 20th century, identifying the systematic variations of micro and macro grooves in recorded performances, for further comparative analysis in diachronic and synchronic perspective of artists and music groups. The Groovemic analysis method is based on the Integrated Analysis Model (IAM), described in Caporaletti [3]which advances in an audiotactile perspective of the Systematic Variations of Duration (SYVAR-D) paradigm described by Bengtsson and Gabrielsson[4] , but also seeks the constant updating and implementation of new protocols such as Swing Ratio [5]. The computer branch of NESCoM offers advice on the computational tools available to extract musical data, transcription by musical notation, as well as theoretical and methodological contributions of musicology (history, theory and musical analysis), ethnomusicology, popular music and Cultural Studies, articulated with the fields of philosophy, social sciences and cognitive sciences, mediatics, and anthropology.

This project is implemented by Jennifer Soares Nogueira and is coordinated by Fabiano Araújo da Costa.

4. Digital Musical Instruments Design

Historically, an HCI perspective is often related to concepts, models, methods and techniques whose final intention is to optimize the user interface (UI) around how people can, want, or need to work, rather than forcing the users to change how they work to accommodate the system or function. In fact, an HCI perspective is based on the needs of the users and, for that, we need to know them, their goals and tasks. In other words, to adopt an HCI perspective in computer music converges towards the central idea that, to design more useful and usable systems, we must better understand users and the tasks they undertake, and better apply our understanding of these tasks in the design process. Such approach (known as User-Centered Design, UCD) is also valuable for the design of digital devices-applications-gadgets for music.

People usually have an "emotional" affection towards their acoustic instruments and they bonded with its character [6]. This is very different in regard to people's feelings about their digital instruments and, a big part of it, is due to the user experience (UX). As the name suggests, UX design is about designing the ideal experience of using a service or product. To achieve this, the users must be in the centre of the designing process (UCD approach): they must not only be listened but also be involved. Overall, it is essential to take into account what people are good and bad at, both in a motor and cognitive level.

Until recently, most scholars were wary of the notion that music cognition could have a biological basis, and this fact reflects on the limited support HCI offers to the design of computer music technology. However, the growing interest from the Computer Music community in digital and computer interfaces used for music performance is a clear indication that Computer Music (CM) researchers have become aware of the importance of HCI studies. Indeed, interfaces for musical performance present immense challenges for HCI, since such interfaces provide the interaction between a performer and a computing system (involving several complex cognitive and motor skills), they make computers turn back toward being things that fit our physical, embodied natures, rather than only operating in the realm of symbolic processing. Therefore, such interfaces for musical performance should be designed around human inputs and outputs rather than computer ones [7] and this claim remains true in the context of computer music. Following, some of the ongoing projects in that particular line of research.

4.1 TIME – Tangible Interface for Musical Education

The project consists in building a tangible interface to test the relation between the psyche (mental processes) and motoric (physical activities) in a musical context aiming to validate some of the cognitive traits of musicality. The proposed interface is a surface, equipped with computer vision, where inexpensive non-tech materials (slime, playdough, clay, ink, etc) are freely shaped and moulded in different directions and forms. The idea is to establish natural mappings between sensorimotor/manipulation skills and sound perception in order to propose better designing guidelines to design new digital musical instruments.

Psychologists and educators have been using such materials as a tool to let the children to express and communicate ideas, thoughts and feelings; It is sensorial tactile experience that stimulate the children to respond creatively to what they see, hear, smell, touch and feel. Since the goal of the project is to gain a deeper knowledge of innate cognitive traits involved in musical activities, it has been decided that the same strategy should be used in the communication with children subjects.

This project is coordinated by Prof. Leandro Costalonga and it has been implemented by Luiz Guilherme Rodrigues Meireles and Carlos Henrique Fernandes Louzada da Fonseca, both students of the Computer Science course (CEUNES/UFES)

4.2 Assistive Technology for Expressive Musical Performance

According to the latest Brazilian census, around 45,6 million Brazilians (23,9% of the overall population) declared some sort of disability. Out of this, 13.1 million are motor impaired people. In theory, there have all sort of laws granting equals opportunities to disable people. The reality is quite different though. Designing interaction for motor impaired people give us the opportunity to apply the musicality-oriented interaction design framework [8]. As a case of study, a performance tool for real-time improvisation based on mashup has been developed. The overall idea is to feed a database with musical data (audio) continuously retrieved from music streaming services, segment it based on a human-like (perceptual) rhythm pattern recognition metric and establishing a classification matching criterion to merge the samples based on the performer's actions.

It is important to realize the complexity of extracting and organizing this enormous amount of date and offers to the performer just the samples that are adequate both aesthetically and technically. This is done applying Music Information Retrieval (MIR) techniques. Equally challenging is to design the interaction between this vast database and the motor impaired performer with all the support that is required in the context of a live musical improvisation. Hence, a study about musical creation using audio segments recombination extracted from WAV files is currently ongoing - something inspired by the process of cutting and pasting of magnetic tapes used by the pioneers electroacoustic music. Some results can be listened at [https://bit.ly/2XHvbm1].

This ongoing research in a master project developed by Higor Camporez and it is supervised by Helder Rocha and Leandro Costalonga.

4.3 Musicality-oriented programming

In recent years, the two sides of the brain have come to symbolize two sides of human nature; the left brain hailed (or disparaged) as "logical," "analytical," and "intellectual," and the "intuitive" right brain as the avatar of emotion and creativity. The ability to produce and respond to music is conventionally ascribed to the right side of the brain [9] Interestingly, Magnusson & Mendieta [6] reported that one of their subjects believed that acoustic instruments tended to be more right brain or spatial, and digital instruments tended to be more left brain and linguistic. Now, if you think about computer programming, there is little doubt it is associated with the left-brain side, therefore, it is safe to assume that music making using computer programming languages does not explore the same cognitive arsenal as that music composed using traditional strategies. There is no judgment regarded to their artistic value or aesthetics here, just an assumption that the current programming languages for music were not designed to consider our inner musicality.

This project aims to propose a programming language that is truly designed to support the intuitive, creative and analytical processes involved in musical activities. For that, a series of studies of the current languages and environments (like SuperCollider, ixilang, Sonic Pi, CSound, Pure-data, etc) is being held with musicians in order to assess its strong and weak points.

This project is coordinated by Prof. Leandro Costalonga and it has been implemented by Herick Lima (Computer Science, CEUNES/UFES)

4.3 Artificial Intelligence applied to Musicality

Csikszentmihalyi [10] argued that an intrinsically rewarding state of consciousness (negentropic state) is achieved when skills and challenges are in equilibrium, and music is basically done for pleasure.

The translation of musical thoughts into actions usually requires, not only talent but also a fair amount of practice. These actions can be either motor-related, such as playing a fast passage on a traditional musical instrument, or cognitive-related, such as structuring and developing a musical algorithm. To amend that, several commercial musical products offer musical libraries composed by "building blocks" (samples, patches, rhythmic patterns, etc.) as well as intelligent assistance features that facilitate the process of creating music. However, the level of the assistance offered must reflect a well-thought balance between aid and intrusiveness, since there is evidence that more knowledgeable individuals are less likely to enjoy it.

The said, this project studies ways to conciliate skills, challenges, and assistance by using AI approach, in order to offer features such as: autocomplete musical phrases, intelligent harmonizers and timbre recommendation system based on a particular musical style, performance-related assistance such as adaptive accompaniment systems or an intelligent "autopilot" mode for live coding.

This project is coordinated by Prof. Leandro Costalonga and it has been implemented by Luis Henrique Gundes Valim (Computer Science, CEUNES/UFES)

5. Final Words

The NESCoM is a relatively new research group, officially founded just over 6 year ago (2013). It is a multidisciplinary group formed by researchers from different departments, including computer science, computer engineering, electrical engineering and, of course, music. It is geographically located at two campi of the Federal University of Espírito Santo, with the Music and Electrical Engineering departments located and the main campus in Vitória-ES (Campus Goiabeiras) and the Computer Science and Computer Engineering at the north campus in São Mateus, just over 220 km away. Unfortunately, it must be acknowledged that the distance is indeed a limiting factor in the interaction and collaboration between these two branches. NESCoM is also a member of the of G-UbiMus, an international multidisciplinary research network composed by UFRGS, UFAC, IFAC, UFSJ, UFES, Maynooth University (Ireland), Linnaeus University (Sweden) e Queen Mary University of London (UK).

Over the years, the group has produced over 35 peer-reviewed papers, 5 book chapters, and a book. In addition, over 40 pieces were composed for concerts, cinema soundtracks, and audio-visual productions. Academically-wise, over 40 monographies and 3 master dissertations were written. In fact, the group is never short of engaged talented students that wish to join in.

At the moment, the group is working to improve collaboration with other research groups with similar research interests at the university, such as the [eMMa] Núcleo de Estudos em Música e Musicologia Audiotátil. Also, an ongoing project involving the Sound and Music Computing Group (University of Padova, IT) and Music Technology Lab of Maynooth University (IR) is looking into guidelines to design more intuitive musical instruments.

References

1. Caporaletti, V.: I modelli cognitivi visivo e audiotattile: criteri epistemologici e modalità di implementazione. 5, 97–123 (2018)
2. Costa, F.A.: ARAÚJO COSTA, F. (2018) Groove and Writing in Radamés Gnattali's Toccata em Ritmo de Samba No 2. RJMA - J. Jazz Audiotactile Music. Stud. English Noteb.
3. Caporaletti, V.: Swing e groove : sui fondamenti estetici delle musiche audiotattili.
4. GABRIELSSON, A., BENGTSSON, I., GABRIELSSON, B.: Performance of musical rhythm in 3/4 and 6/8 meter. Scand. J. Psychol. 24, 193–213 (1983)
5. ELLIS, M.C.: AN ANALYSIS OF "SWING" SUBDIVISION AND ASYNCHRONIZATION IN THREE JAZZ SAXOPHONISTS. Percept. Mot. Skills. 73, 707 (1991)
6. Magnusson, T., Mendieta, E.H.: The acoustic, the digital and the body. In: Proceedings of the 7th international conference on New interfaces for musical expression - NIME '07. p. 94. ACM Press, New York, New York, USA (2007)
7. Miranda, E.R., Wanderley, M.M., Kirk, R.: New digital musical instruments : control and interaction beyond the keyboard.
8. Araújo Fim Camporez, H., Frizera Neto, A., Lesqueves Costalonga, L., Roberto de Oliveira Rocha, H.: Interface Computacional para Controle Musical Utilizando os Movimentos dos Olhos. (2018)
9. Weaver, E.: A more holistic picture.
10. Csikszentmihalyi, M.: Imagining the self: An evolutionary excursion. Poetics. 21, 153–167 (1992)

Computer Music research at FEEC/Unicamp: a snapshot of 2019

Tiago Fernandes Tavares[1]*and **Bruno Sanches Masiero**[1]†

[1]School of Electric and Computer Engineering – University of Campinas
Av. Albert Einstein, 400 – 13083-852 – Campinas, SP

tavares@dca.fee.unicamp.br

Abstract. *This is a lab report paper about the state of affairs in the computer music research group at the School of Electrical and Computer Engineering of the University of Campinas (FEEC/Unicamp). This report discusses the people involved in the group, the efforts in teaching and the current research work performed. Last, it provides some discussions on the lessons learned from the past few years and some pointers for future work.*

1. Introduction

The School of Electrical and Computer Engineering (FEEC) of the University of Campinas (Campinas) is located in the district of Barão Geraldo, in Campinas-SP, Brazil. It offers two undergraduate-level courses, namely Electrical Engineering (five years, with 70 students per year) and Computer Engineering (five years, with 90 students per year). Also, it offers graduate-level courses in Electrical Engineering, both in the MsC (two years) and PhD (four years) levels.

Computer music research at FEEC builds upon the existing Engineering *syllabi* as discussed in Section 2. As a consequence, a large amount of our research work draws from Engineering perspectives. Nevertheless, we have learned important lessons on fostering multidisciplinarity in computer music research. Insights on this topic are presented in Section 3.

2. Teaching

The computer music group at FEEC deeply relates to the Electrical Engineering (EE) and Computer Engineering (CE) courses. As shown in Table 1, both courses provide students with hours in three relevant topics: computer programming, digital signal processing, electronics, and embedded systems. Additionally, CE students are provided with 60h courses in artificial intelligence.

Table 1: Regular course hours in EE and CC related to relevant topics for computer music.

Topic	E.E.	C.E.
Computer Programming	60h	360h
Digital Signal Processing	240h	240h
Electronics and Embedded Systems	280h	200h
Artificial Intelligence	0h	60h

This background tends to direct research towards technical perspectives on computer music problems. As such, we can highlight the following problems:

1. Build Music Information Retrieval (MIR) systems and evaluate them using objective (Accuracy, Recall, Precision) measures,
2. Explore the impact of features in label prediction experiments,
3. Build digital musical instruments and digital musical interfaces based on electronic sensors and microcontrollers.

This naturally points our research to the interests of the NIME and ISMIR communities.

Although the regular courses yield a strong technical background, some other skills are only available in optional (elective) courses. These skills, shown in Table 2, are important to computer music, but are not currently present in the EE and EC courses.

Table 2: Computer music related skills that are currently missing in EE and EC courses.

Skills
Analyze musical pieces (aesthetically and culturally)
Perform qualitative analysis in audio results
Synthesize audio

Such skills can be employed in computer music research both to interpret results and to draw insight from artistic roots. Also, such skills favor the understanding of the meaning of results and can broaden the reach of the technical research. In special, they can foster the possibilities to:

1. Interpret the results of MIR systems under the light of artistic and cultural domain knowledge,
2. Draw insight on the sound characteristics that relate to specific features for further exploration,
3. Integrate synthesis and performance into the process of building digital musical instruments.

Although these skills can be acquired in elective courses within Unicamp, we decided that a more focused approach would be necessary. For such, we offer a series of advanced study courses that compose a study certificate in Sound Engineering.

2.1. Study Certificate: Sound Engineering

The study certificates at Unicamp began as an initiative to recognize the efforts of undergraduate-level students that acquired advanced knowledge. FEEC currently offers six different certificates, and the most active ones are in Biomedical Engineering and in Sound Engineering.

To earn a certificate in Sound Engineering, a student must receive credits related to four different courses,

*Supported by CNPq.
†Supported by CNPq.

as shown in Table 3, as well as perform at least one semester of research in an audio-related field. The courses within the certificate aim at enabling students to perform research in computer music and in acoustics, which are two related fields that have active groups at FEEC.

Table 3: Computer music related skills that are currently missing in EE and EC courses.

Course	Hours	Content
Introduction to Sound Engineering	30h	Series of lectures with invited speakers aiming at providing an overview of the methods and current research in acoustics and computer music.
Sound Engineering 1	60h	Acoustics and electroacoustics. Modelling of loudspeakers, room, and environmental acoustics.
Sound Engineering 2	60h	Analysis and synthesis of audio signals. DSP and machine learning techniques. Audio synthesis using diverse techniques.
Elective	60h	Any audio-related course outside of FEEC

These courses were specifically designed to fit within the EE and CE *curricula*. As such, they have prerequisites in computer programming and in advanced calculus. This means that, although non-engineers can earn the certificate, they will have to go through all the base courses for such.

This is an impedance, but we have provided a complete guide on how to organize the workload throughout the courses. Currently, there are many students from the Music School that are taking calculus and programming courses in order to earn the certificate.

Also, we highlight that the Sound Engineering 1 and 2 courses are jointly offered as graduate-level courses. As a consequence, students that earn the certificate also earn credits that can be used in a MsC-level degree. This facilitates them to pursuit research in the field.

3. Discussion on Multidisciplinarity

The computer music research group started in 2015 at FEEC, and some important lessons have been learned. The main difficulty at this point has been successfully fostering multidisciplinarity, because this requires time and adequate environments.

Multidisciplinarity is a perspective, that is, a possible way of looking at problems. As such, computer music should not be seen as a set of engineering problems nor as a set of artistic problems.

We believe that fostering multidisciplinarity involves a change of perspective. This requires understanding one's background perspective. This means that before approaching multidisciplinarity students should be able to understand why they look at things the way they do.

Two important perspectives that are present within the university are the archetypical artistic and technical perspectives. The archetypical artistic perspective involves seeing things regarding their aesthetics and cultural impact, whereas the archetypical engineering perspective relates to a point of view in which problems can always be objectively defined, solved, and evaluated. Both perspectives are important in computer music, because CM-related research draws both from cultural/aesthetic discussions and from technical aspects to derive tools and solutions. However, it is very common that only one of these perspectives is reinforced in specific courses.

As a consequence, participating in computer music research requires a personal construction of conscience regarding one's own perspective. Also, it requires experimenting with other perspectives, as to build an embodied experience of them. This transcends the classroom possibilities, and requires an environment for free creation.

For such, we built a computer music electronics lab, in which computer music experiments can be performed. The laboratory has a standard electronics workbench, and is annexed to a recording studio. Currently, it is used to develop sensors, robots and other creative, CM-related artifacts. The users come from both the music and the engineering backgrounds, and they intentionally interact and learn from each other.

We still have not had science throughput from this specific interaction. This depends on further work, which is bounded to regular university schedule constrains.

Nevertheless, the participating students' progress is visible and the learning of technical skills has been accompanied by the learning of new perspectives by each one of them. We are glad to foster this environment and make it easier to embrace all different perspectives that compose our University.

ALICE: Arts Lab in Interfaces, Computers, and Everything Else - Research report (2019)

Flávio Luiz Schiavoni, André Lucas Nascimento Gomes, Avner Maximiliano de Paulo, Carlos Eduardo Oliveira de Souza, Fábio dos Passos Carvalho, Frederico Ribeiro Resende, Gabriel Lopes Rocha, Igino de Oliveira Silva Junior, Igor Alves Estevam Andrade, João Teixeira Araújo, Luísa Lugon-Moulin Brazil Santos, Luan Luiz Gonçalves, Mauro César Fachina Canito, Octávio Ferreira Deluchi Cerqueira da Silva, Rômulo Augusto Vieira Costa, Thiago de Andrade Morandi, Valdir de Siqueira Santos Junior

[1] Arts Lab in Interfaces, Computers, and Everything Else - ALICE
Federal University of São João del-Rei - UFSJ
São João del-Rei - MG

fls@ufsj.edu.br

Abstract. *Located in the Computer Science Department of the Federal University of São João del-Rei, but not limited to just this space, ALICE has emerged as a research group focused on the development of software, art and technologies for the area of computer music and digital arts. Over the time, ALICE became the laboratory to develop technologies to an artistic group, called Orchidea, focused on the creation of digital art, encompassing students from diverse areas, such as Computer Science, Scenic Arts, Architecture and Music in a transdisciplinary context of art creation. In this way, this work aims to present the various initiatives and proposals carried out by the ALICE, addressing the development of technological products, through the tools implemented and the external tools used for teaching and for artistic creation. In addition, this work describes the current researches that are under development by the members of the groups, also highlighting the development of the digital performance titled "O Chaos das 5" and all the aspects and learning that we have obtained so far with this performance.*

1 Introduction

ALICE[1] is the name of a laboratory in the Computer Science Department at the Federal University of São João del-Rei. The name of the lab is an achronimous to Arts Lab in Interfaces, Computers, and (Education, Exceptions, Experiences, Entertainment, Environment, Entropy, Errors, Everything, Else and Etcetera...). This name is also a good hint to understand some concepts about the lab.

Firstly, a direct reference to Lewis Carroll book, Alice in the wonderland, can bring the idea of some nonsense, fantasy and peculiar ideas that could guide researches on the lab. Lewis Carroll, a mathematician, used logic to entertain child and to create art. Certainly it is not a "really serious" lab and we believe that it is possible to have interesting computer science involving, mathematics, arts and humanities. Secondly, it is an Art Lab. The presence of arts in the name can be another good hint to understand that we are worried about creative process and some ephemerous software creation, group performance and so on. We are also worried about interfaces and how the man can interact with the world, specially when involving computers to perform this interaction. Thirdly, the infinity possibilities of an open name, that includes everything else, can really be a good chance to keep an open mind when thinking about computer science and art research.

Most part of the members of this lab are computer science undergraduate students, studying computer science. There are also computer science master's students (PPGCC - *Programa de Pós Graduação em Ciência da Computação*) and master's students in Arts, Urbanities and Sustainability (PIPAUS - *Programa Interdepartamental de Pós-graduação Interdisciplinar em Artes, Urbanidades e Sustentabilidade*). Since ALICE started to be an laboratory to art creation, some students in Music, philosophy, scenic arts, applied arts and also professors from different areas joined us in some projects.

In this paper, written by several hands, we will present the current research, art creation and software developed in this lab.

2 Teaching and learning - The STEAM

Since our University has no course on Digital Arts, our students has a few options of courses to get involved in this subject. There is only one course, Introduction to Computer Music, available to undergrad students into Computer Science. To Master's students there are two courses, Digital Arts and Immersive performance in Scenic Arts. Thus, the get involved in the areas of digital art, it is necessary some complementary studies.

Inspired by the concepts of the STEAM framework [1], ALICE supports a group that bring discussions from different points of view in the STEAM areas. STEAM is basically an educational system where through art, students are introduced to concepts related to Science, Technology, Engineering and Mathematics.

Regarding the issues studied within the study group, it was discussed how some mathematical definitions, such as Symmetries, Fractals, Fibonacci numbers, among others, could be applied through several areas such as Computer Science, Music, Architecture, Photography and Applied Arts, all with their respective points of view,

[1] https://alice.dcomp.ufsj.edu.br/.

employing these concepts under various types of media, such as images, videos, and audios.

By having a heterogeneous group of members, diverse departments and with a completely different experience and formation brings a collaborative and immersive environment where it becomes a place the members acquire an evolution of their work under the influence of other factors, apart from their experience.

3 Art Creation - The Orchidea

To try to reach some art creation, we also created an art group, called Orchidea. The Orchidea - an Orchestra of Ideas, focuses on the development of collaborative art and the creation of an environment that encompasses students from diverse areas, be it theater, music, computer science, architecture, philosophy and others, to create art together. By approaching this transdisciplinary concept, Orchidea ends up being a very promising environment in relation to the collective learning of the students, since students from different areas, have different abilities, which in group, can be shared, making the students help each other among themselves, overcoming their difficulties.

Initially, the idea of the name of the group appeared in 2015, with the implementation of the tool called Orchidea [2]. This tool was a mobile application, implemented in Java, where its purpose was to enable the creation of a mobile orchestra. The application was developed by an undergraduate student, who at the end of his monograph, ended up giving no continuity to the project.

In 2016, as a final result of the discipline of Introduction to Computer Music, Orchidea was resumed, gaining a second version, now based on the Web Audio API and mobile device sensors. Finally, in 2018, based on the need to create art and study new aesthetic concepts associated with technologies, our digital art group was finally created.

Currently, Orchidea has undergraduate and graduate students, and its objectives can be described as:

- Fostering and performing digital artistic creations;
- Integrate transdisciplinary knowledge of different areas through art;
- Stimulate the collective and collaborative creation of art supported by the computer;
- Use affordable and more sustainable technology for artistic creation;
- Use and create free software for the diffusion of artistic creation and Enable public participation in the presentation and artistic creation that can happen anytime, anywhere.

3.1 Current Art Creation - O Chaos das 5

As a more significant result developed by Orchidea, one can highlight the digital performance called “O Chaos das 5”, already presented in São Paulo, Belo Horizonte, Conselheiro Lafaiete, Mariana and São João del-Rei. This performance is a spectacle that involves the use of technology, besides providing the interaction of the public, which ends up taking a more active position during the play, influencing it and becoming an artist and not just part of the public.

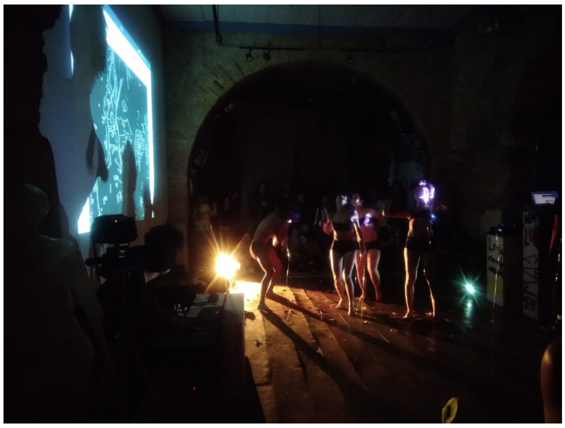


Figure 1: Digital Performance “O Chaos das 5”

The performance was developed in the city of São João del Rei in 2018, with the support of the ALICE laboratory, in conjunction with the ECOLAB / Movére research laboratory of the Performing Arts course. In “Chaos das 5” technology influences audience experience and is influenced by audience interaction through mobile devices, where the whole plot is given by three layers: visual layer; gestural layer and musical layer.

The visual layer is based in images created in real time, projected on the walls, presenting the aesthetics of placing the computer in the scene, aiming at the idea of opening the “Black Box”, showing how the work is developed and not just its respective end result [3, 4]. The gestural layer has a performance that mixes scores of gestures and improvisations in an immersive theater that allows physical interaction with the audience in the space of the show, encompassing artists who interact with the audience during the performance. Finally, the musical layer with DMIs to be accessed by the public by their mobile devices, in addition to having musicians improvising with their unconventional digital instruments [5, 6].

In relation to the tools developed for the performance, it is worth mentioning the creation of DMIs for the audience, DMIs for our musicians, visual effects, image processing applications, image synthesis, Pure Data patches, web audio instruments and more. All the technology used in this performance was made in ALICE and is the outcome of a student’s research.

4 Current Research

Currently, our group has carried out several researches in several areas related to Computer Music / Digital Art. This subsection will present the work done by ALICE, giving a brief explanation of what has been developed, in addition to the ongoing research.

4.1 Digital Musical Interfaces

The creation and use of Digital Music Instrument (DMI) is one of the lines of research that is currently being developed in our group. Unlike an acoustic instrument, where the creation of an instrument goes through a long period of trial and error, a digital instrument has as its characteristic that in its conception process, the developer looks for achieve a certain goal [7].

This research aims at studying the creation of DMIs from ubiquitous devices, raising several implementation aspects, such as which forms of mapping will be implemented. Mapping is a nice subject to explore when creating new instruments since they can be from the simplest, one-to-one, where a button is responsible for to play only one note, to the more complex ones, many-to-many, where a sequence of buttons can perform certain actions, which in turn can also influence future actions. It is worth mentioning that all the design decisions raised during the development of an DMI are studied as well as the considerations on how the creative process can influence and be influenced by the technological decisions used.

Figure 2: DMI development based on the joystick

For instrument development, we must work under the construction of three modules: Firstly, one must obtain or produce an iteration device, so that it is an interface where it is usual and interesting for the musician to perform on it. Examples of elements used as interface are buttons and knobs of MIDI devices and Joysticks [8], the capture of movements through cameras [7], smartphones [9] and their sensors, among others.

Another tool that has to be used is an apparatus that synthesizes the media that the instrument reproduces. In addition to reproducing the media thought by the creator of the instrument, via images, sounds, actuators and others, it must be able to return feedback to the user of the instrument, so that it can perform an action from this feedback.

Finally, the composer of the instrument must interconnect the two previous elements via a mapping module. These combinations of connections made by the creator will define how it will work and control the component it interacts on the synthesizer.

In our laboratory, we had several initiatives of DMI development, like a Joystick do play music (presented in Figure 2 or web browser based instruments (Figure 3).

4.2 WebArt

Since the early days of computing artists have been using computers as tools to support their traditional artistic creations. This includes the various areas of art such as photography, painting, theater, dance, music, architecture, sculpture and literature, for example, domains that have been radically impacted by the emergence of computational support. Over time, the computer has gone from being a tool to support the traditional arts (from the so-called Fine Arts to the so-called “Minor Arts” as the craft) for a specific creation tool and not tied to the conventional arts. This allowed the emergence of another type of art, possible only through electronic devices, called Digital Art. Digital Art emerges as a convergence between Art, Science and Technology and has brought artists new challenges for artistic creation in the digital universe [10]. This Digital Art includes several new forms such as Media Art, net art, digital installations and Virtual Reality as a form of Art [11].

However, with the emergence of the Internet, the web started to arouse the interest of several areas, including art. The Internet was initially used as a way of distributing traditional art, through shops, museums and virtual art galleries, and also as a way of making art by allowing multimedia creation and being present at the center in the technological revolution of information [12]. This art form, called Web Art, includes several artistic creations made especially for the web [13].

Thus Web Art can be characterized as a category of digital art that only occurs for and through the internet, being worked in the web environment has a huge range of possibility, yet has as its main feature the interactivity, which gives the user the possibility of interacting with your content, in a scheduled or real-time manner.

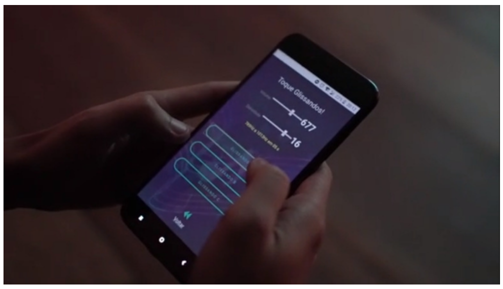


Figure 3: Web based DMIs developed with Mosaicode to the performance O Chaos das 5

In our laboratory, several initiatives using web art were developed [14, 15, 16, 17, 18]. From web based DMIs (Figure 3) to visual arts (Figure 4) and video collage performance. Our group also conducts research to explore the principle of digital art and how it, developed to work with any electronic devices, evolved into web art, tools available in HTML for the development of web sites aimed specifically at the artistic making, which differ from the traditional pages developed only from artistic purposes. In this way, discussions about the beginning of the web art are carried out, addressing how were the environments of creation of the first web arts, what are the resources available in the initial versions of HTML, besides what are the new tools and new possibilities of web creation art that HTML5 brought to the surface with its emergence.

4.3 Creative process and collaboration in art creation

The term "creative process" characterizes the methods used by creators in the composition of their artistic products, and this action is performed in any subdivision of the art. It is an action of semantic and semiotic translation, where the whole process of artistic production is based on a repertoire, be it of memories, concepts or other subjectivities. This first sums up a second repertoire of material elements such as colors, words, sounds and forms, within a creative phenomenological instance, which will be the creative and communicational selection screen that will conceive the final artistic product [19].

In short, the creative process can be associated with Dewey's philosophy, which treats the artistic method as an organic process, which consists of a trajectory from gestation to maturation. Therefore, the work of art is not inseparable from its creative process, since the games of trial and success, waiting and discovery, between the stages of idealization, execution and conception, all of these compose temporal instants which resulted in the finished work of art [20].

Uncharacterized from a set of rules that govern their formation, the artistic creative process has only one premise to be followed, that "the rule of art is the criterion of success" [20]. In other words, the process of artistic creation is limited only by the very rule that art imposes on it to conceive itself, so dualities such as rigor and spontaneity are not expressed as absolute rules, where only the inventiveness and originality of the artist should flourish.

In ALICE, the creative process is a subject always explored, specially the collective creation, cooperative creation and the use of technology to aid creative process in a group context [21, 6, 22, 23, 24, 25]. We also strongly believe that it is possible to attach software development to art creation and understand the creativity to develop computer algorithms as a special case of art creativity.

4.4 Computer Music Programming Languages

The area of computer music is strongly based on the generation of audio via computer, which is usually created under a software system, which in turn is developed under software development methodologies that encompass programming, from programming languages, that are directed to the expression of audio synthesis and processing, culminating in the emergence of several non-conventional computational solutions. In relation to the programming languages used to perform any computational task, they are under constant influence of evolution and innovation, and these languages are the focus of some theoretical and practical research developed on ALICE.

An example of these theoretical research is a general study of different languages with different paradigms, in the Computer Music domain, addressing the historical questions of the evolution of these languages, their technical and developmental issues, taking into account ease of use and criteria such as readability, expressiveness and ease of writing [26, 27].

Some of the theoretical research on musical programming languages is used to help the development of the Mosaicode environment, a visual programming environment for code generation of Domain-Specific Languages (DSLs), that aims to help novices to create applications in Digital Art areas such as Computer Music, Computer Vision and Computer Graphics.

4.5 Audience Participation and interaction in Digital Performance

The research in the area of digital performances includes musical performances and theatrical shows that involve various aesthetic concepts related to some kind of technology for their execution, whether they are sensors, cell phones, or some another technological resource. Through an in-depth study of the digital performances until then, this research aims to understand how the public interaction in these events is given, raising aspects about what should and should not be done so that the gap between the definition of who is public and who is an artist becomes more and more narrow, without the engagement of the participants being compromised.

Still on digital performances, the audience's interaction can be done in several ways. In our lab we are exploring the audience participation via mobile phones, which are used in the form of DMIs, enabling the public to actively participate in the performance by generating music from mobile devices [25, 14, 9, 28, 24].

4.6 Art and Sustainability

Another area of research addressed by the group is art and sustainability, which looks for use computational arts in the development of a sustainable consciousness based on the Brundtland Report [29]. Among the issues addressed in this report, we can highlight the reduction of energy consumption, the development of technologies using renewable energy sources and the increase of industrial production in non-industrialized countries based on ecologically adapted technologies, which are directly linked to the computational area.

In ALICE, the sustainability art and technologies follow the ideas of sustainability involving social, econom-

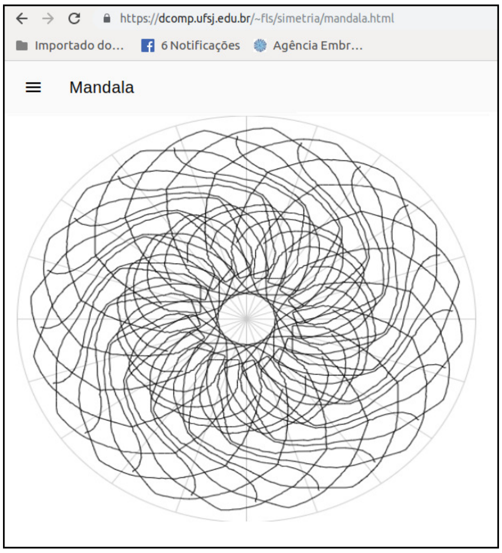


Figure 4: Web based interface that explores symmetry in visual arts

ical, cultural and environmental sustainability [30, 31]. Initially, it is worth mentioning that in ALICE only free software are used and developed [32]. All tools used in our group are open, modular and based on code reuse. In order to ensure greater compatibility, in addition to ensuring aesthetic possibilities, ALICE works with open protocols, file formats and systems compatible with legacy systems and other platforms. In this way, we believe that we are not only creating new technological tools, but we are also allowing others to use them to create art or even to learn computing.

The usage of open source tools, open hardware, open source libraries and all kind of open technology is mandatory. Also, the usage of ubiquitous devices that do not demand expensive technology is also a guideline. Thus, this interaction of the public with sustainable artistic production would to potentiate the development of an awareness based on the Brundtland Report, making sustainable computer art reach its goal.

Another concern focusing sustainability in our lab is about how to combine new technology with old culture from our land. The research involving cultural sustainability led us to bridge cultural aspects of our society with digital arts, like in our soundscapes projects involving church bells and ubiquitous technology [33, 22, 23, 34, 35].

5 Software development focused on Arts

Beyond theoretical researches and art creation, ALICE has developed a few technologies. The most part of the software developed in our lab are only prototypes and ephemeral code but some of them should be mentioned here.

5.1 Mosaicode

Mosaicode is a visual programming environment geared towards the Digital Art domain [36, 37]. It works from blocks, which are elements with specific and atomic functionalities on a domain, and connections, which connect these blocks and perform data transmission between them. Connections and blocks together form a diagram, which is an abstraction of an application. With this artifact, it is possible to generate, execute or edit the code of an application. This capability allows the environment to be a powerful software for programming, as well as enabling rapid prototyping and coding. The tool, besides being open-source, is extensible. Each extension allows the user to be able to use concepts related to the domain in which the extension belongs. If it is in the user's interest, the environment provides the same viability to freely create its own extension, representing the domain of its purpose.

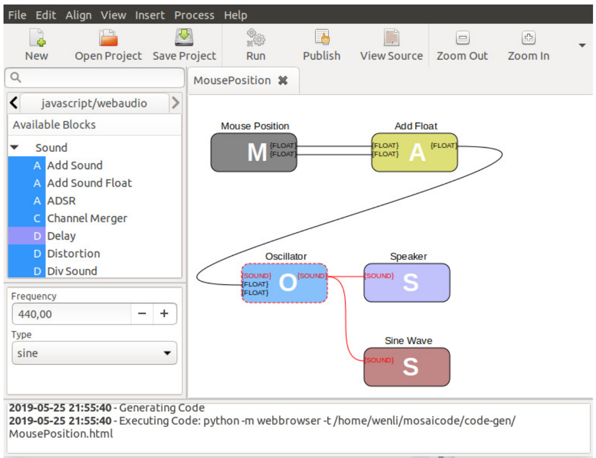


Figure 5: Visual Programming Environment Mosaicode

The tool emerged in 2014, based on the refactoring of Harpia, a tool for training and management of computer vision systems, created by *S2i* in the Federal University of Santa Catarina (UFSC), where its architecture was completely altered and the code completely rewritten to use extensions and plugins.

The environment has applications for teaching [38], digital art creation, software prototyping and more. Most part of the instruments and visual creations used in the performance "O Chaos das 5 " were created using Mosaicode.

Mosaicode to Image Processing & Computer Vision

One of the extensions already developed for the Mosaicode environment deals with the domain of Digital Image Processing and Computer Vision [39]. It provides about 65 distinct features addressing both named areas, using OpenCV framework in C++ language, acquiring computational efficiency in the generated code.

Among these functionalities, there are blocks that portray mathematical, logical, morphological, data generation and synthesis, low-pass and high-pass filters, pattern detection and a variety of other algorithms, and operations that address signal processing techniques in images.

Mosaicode to Image Synthesis

In development process since 2017, the Extension of Image Synthesis of Mosaicode was developed under the OpenGL graphics library under the C / C++ programming language [39].

Currently the extension has 44 blocks distributed in 8 categories: **2D Shapes** and **3D Shapes**, with two-dimensional and three-dimensional models, respectively; **Artistic**, where they use representations of mathematical definitions discussed in the STEAM study group as fractals and symmetries abstracted in blocks; **I / O**, where it has control blocks for the generated application; **Math**, mathematical operations on top of float values; **Operations**, to perform geometric transformations under the rendered forms; **Types**, which declare variables to initialize other blocks and **Window**, where parameters of the created window are modified. To make the connection between these blocks, 3 types of ports have been developed: **Color**, **Float** and **Flow**.

Mosaicode to Audio Processing

Developed in C/C++ and based on PortAudio API and libsoundfile, there is an extension to Mosaicode to develop audio applications, focused on providing resources for working with sound design. This extensions includes blocks to DSP processing, like filters, audio sources, like noise and oscillators, audio input, output and blocks to open / save audio files.

Mosaicode to Web Art

The Mosaicode also has an extension to create Web Art based on the HTML5 APIs [40, 41]. This extension allows to create audio synthesizers using the JavaScript / Web Audio API, which is also geared towards Web Art development. This extension has blocks to play audio files, access microphone, create oscillators, noises and more. Among the various categories of Blocks implemented, we have until then: Sound Sources; Audio Filters; Logical and Arithmetic Operations; and Input Devices.

5.2 Glue Tube

GlueTube is a tool capable of manipulate YouTube videos through an HTML5 API to generate a montage with the videos being shown on the computer screen. The arrangement of the videos can be controlled by the settings on the project and the tool also allows to create a *.XML* file that works like a score, to indicate the youtube video parts and how many times it should be executed.It is also possible to indicate the position on the screen and the size of the video.

Despite the possibility of creating a score, it is possible to mix the videos in real time, like in a musical improvisation performance, and use the score like a backing track to improvise over that.

The conception of this tool was based on Pierre Schaeffer's concrete music and the collage art, a technique which is very popular in the web art, applied to the YouTube collection. However, in order to follow YouTube's copyrights rules, the infringement has been considered and we created a tool capable to use fragments of YouTube videos, online, with no necessity of downloading them.

5.3 Ha Dou Ken Music DMI

As discussed in subsection 4.1, the creation of DMIs is a very present area in our research lab. One of the works related to the creation of a musical instrument is the "Ha Dou Ken Music DMI" [7, 8, 42].

This work aims to use video game controls as an interface for musical interaction, taking into account its great presence in popular culture and its ease of access for low prices. Even people who are not in the habit of playing electronic games have possibly already interacted, at some point, with that kind of control. Interactions like pressing a sequence of buttons, pressing them simultaneously, or sliding the fingers through the control, are often present in fighting games like Street Fighter and are used in this instrument to create music.

In this way, the main objective of this instrument is the elaboration of a strategy in which several gestures captured by the interface can influence one or even several sound parameters. That is, the challenge lies in the elaboration of several mapping techniques, be they the simplest, such as the one-to-one mapping where a button causes a certain result that is always static, to the more complex ones, such as the many-to-many mapping, exploring greater expressive video game control capabilities from capturing complex moves. These mappings are performed based on the MIDI protocol, which give us access to a large number of compatible synthesizers.

It is worth emphasizing that experiments show that this choice is capable of influencing the musical expression made possible by a DMI, which makes it closer to a "real instrument", being in this way more pleasurable and challenging to play. This instrument is used in the performance "O Chaos das 5".

6 Final Considerations

From the course of Introduction to Computer music, taught in 2014 at the Federal University of São João del-Rei del-Rei, and also from the interest of some students to continue their studies in the area of Computer Music and Digital Arts, was created ALICE.

Since then, this laboratory has worked extensively on research aimed at technological creations, covering several areas, among them Computer Science itself and also Music, Architecture, visual arts and Theater. This transdisciplinary character brought us positive aspects, since people from different areas have different abilities and ways of thinking. The difference between the students, when analyzed under the collaborative knowledge provided, is quite promising because the students can help each other, making the difficulties of each one overcome from the support provided by the group in general.

Besides the research developed by ALICE, we have the development of software for the creation of digital art. This development provided the opportunity to the students involved with it to think beyond the code while developing a software. Several times it was necessary to think artistically, carrying out a study of the aesthetics that involves the creation of an artistic application, thinking in the creation of digital art and not only in technological tools, accepting borders and limitations, striking with latency and performance issues, trying to not think only in developing codes but in the creation of an art piece.

Computer students generally think only as scientists and almost never as artists. From the creation of the Orchidea group, which is also a consequence of the creation of ALICE, we have been able to break this barrier. Coming in contact with people and working in other areas, all the students involved have changed greatly from the development of the main work of Orchidea, the digital performance "The Chaos das 5". This performance brought with it some technical issues and difficulties that until them many of those involved had not witnessed. Among the problems we have: computational situations, such as the treatment of scalability in high-scale performances and the development of DMIs used; and problems related to the production / execution of a performance, where those involved need to deal with the production of a script that involves references that are not common in a Computer Science course, the allocation of the space to be used, in addition to determining which types of interactions would or would not engage the participating public, and more and more.

Finally, the creation of a research group in the academic environment has much to offer in the development and growth of all people involved, both as students and individuals. It is noteworthy that in the academic environment, specifically in a Computer Science Department, there is a certain difficulty / barrier to creating a group of transdisciplinary digital art, perhaps because of bureaucratic or even cultural issues. However, the creation of ALICE / Orchidea / STEAM has shown us the potential of collaborative work, since we could observe the evolution and development of skills in several students, writing / presenting scientific articles and even improving the ability to deal with people in general.

7 Acknowledgments

Authors would like to thanks to all ALICE members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq and FAPEMIG.

References

[1] Lisa G Catterall. A brief history of stem and steam from an inadvertent insider. *The STEAM Journal*, 3(1):5, 2017.

[2] Flávio Luiz Schiavoni and Paulo Gabriel Nunes Cançado. Orchidea: Os meta-instrumentos da orquestra de ideias. *Revista Vórtex*, 6:1–18, 2018.

[3] João Teixeira Araújo, André Lucas Nascimento Gomes, Frederico Ribeiro Resende, and Flávio Luiz Schiavoni. As poéticas visuais do "o chaos das 5". In *Anais da 2ª Semana Acadêmica Integrada dos Cursos de Letras / 15ª Semana Acadêmica do Curso de Letras*, pages 1–1, São João del-Rei, 2019.

[4] Valdir de Siqueira Santos Junior and Flávio Luiz Schiavoni. Quando a rua entra em cena: uma apropriação do discurso do pixo na performance "o chaos das 5". In *Anais da 2ª Semana Acadêmica Integrada dos Cursos de Letras / 15ª Semana Acadêmica do Curso de Letras*, pages 1–1, São João del-Rei, 2019.

[5] Carlos Eduardo Oliveira de Souza and Flávio Luiz Schiavoni. O discurso da guitarra no espetáculo "o chaos das 5". In *Anais da 2ª Semana Acadêmica Integrada dos Cursos de Letras / 15ª Semana Acadêmica do Curso de Letras*, pages 1–1, São João del-Rei, 2019.

[6] Octávio Ferreira Deluchi Cerqueira da Silva and Flávio Luiz Schiavoni. Diálogo sonoro em chaos das 5, das ruas ao concreto e sintético. In *Anais da 2ª Semana Acadêmica Integrada dos Cursos de Letras / 15ª Semana Acadêmica do Curso de Letras*, pages 1–1, São João del-Rei, 2019.

[7] Gabriel Lopes Rocha, Avner Maximiliano Paulo, João Pedro Moreira, and Flávio Luiz Schiavoni. Desenvolvimento de Instrumentos Musicais Digitais a partir de dispositivos ubíquos. In *Proceedings of the VIII Workshop on Ubiquitous Music (UBIMUS)*, volume 8, pages 98–110, São João del-Rei - MG - Brazil, 2018.

[8] Gabriel Lopes Rocha, João Teixera Araújo, and Flávio Luiz Schiavoni. Ha dou ken music: Different mappings to play music with joysticks. In Marcelo Queiroz and Anna Xambó Sedó, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 77–78, Porto Alegre, Brazil, June 2019. UFRGS.

[9] João Teixeira Araújo and Flávio Luiz Schiavoni. A participação do público em performances artísticas por meio do uso de dispositivos móveis. In *Anais do 5o. Seminário de Artes Digitais 2019*, pages 203–211, Belo Horizonte, 2019.

[10] Oliver Grau. *Virtual Art: from illusion to immersion*. MIT press, 2003.

[11] Bruce Wands. *Art of the Digital Age*. Thames & Hudson, 2007.

[12] Mark Tribe, Reena Jana, and Uta Grosenick. *New media art*. Taschen London and Cologne, 2006.

[13] Annette Weintraub. Art on the web, the web as art. *Commun. ACM*, 40(10):97–102, October 1997.

[14] Igino de Oliveira Silva Júnior, Flávio Luiz Schiavoni, Adilson Roberto Siqueira, and Marcela Alves de Almeida. A interatividade do chaos das 5. In *Anais do 5o. Seminário de Artes Digitais 2019*, pages 115–122, Belo Horizonte, 2019.

[15] Avner Maximiliano de Paulo and Flávio Luiz Schiavoni. A web arte: de linguagens de programação à linguagens artísticas. In *Anais da 2ª Semana Acadêmica Integrada dos Cursos de Letras / 15ª Semana Acadêmica do Curso de Letras*, pages 1–1, São João del-Rei, 2019.

[16] Flávio Luiz Schiavoni, Luan Luiz Gonçalves, and José Mauro Silva Sandy. Mosaicode and the visual programming of web application for music and multimedia. *Música Hodie*, 18:132–146, 2018.

[17] Roberto Piassi Passos Bodo and Flávio Luiz Schiavoni. A sonification approach to music visualization. In *Interna-*

tional Conference of Students of Systematic Musicology, pages 52–57, Belo Horizonte - MG - Brazil, 2018.

[18] Roberto Piassi Passos Bodo and Flávio Luiz Schiavoni. Synesthesia add-on: a tool for html sonification. In *Proceedings of the 16th Brazilian Symposium on Computer Music*, pages 75–80, São Paulo - SP - Brazil, 2017.

[19] Max Bense and Elisabeth Walter. La semiótica. 1975.

[20] Luigi Pareyson and Maria Helena Nery Garcez. *Os problemas da estética*. Martins Fontes, São Paulo, 1997.

[21] Thiago de Andrade Morandi, Flávio Luiz Schiavoni, and Zandra Coelho de Miranda Santos. AnÁlises de processos criativos influenciados pelo espaÇo e memÓria. *Art & Sensorium*, 5:55–68, 2018.

[22] Thiago de Andrade Morandi and Flávio Luiz Schiavoni. Processos criativos de foley, soundscape e música ubíqua em produções audiovisuais. In *Anais do Jornada Interdisciplinar de Som e Música no Audiovisual*, pages 1–1, Rio de Janeiro, 2019.

[23] Thiago de Andrade Morandi and Flávio Luiz Schiavoni. EstÉtica relacional e registro ubÍquo de paisagens urbanas nos processos criativos. In *Anais 5^o Colóquio Ibero-Americano: paisagem cultural, patrimônio e projeto*, pages 1–2, Belo Horizonte, 2019.

[24] Flávio Luiz Schiavoni, Eduardo Silva Xavier, and Paulo Gabriel Nunes Cançado. (Orchidea) Uma primeira aplicação para práticas musicais coletivas na Orquestra de Ideias. In *Proceedings of the VIII Workshop on Ubiquitous Music (UBIMUS)*, volume 8, pages 120–131, São João del-Rei - MG - Brazil, 2018.

[25] Flávio Luiz Schiavoni, Pedro H. de Faria, and Jônatas Manzolli. Interaction and collaboration in computer music using computer networks: An ubimus perspective. *Journal of New Music Research*, 0(0):1–15, 2019.

[26] Rodrigo Ramos Araújo, Flávio Luiz Schiavoni, José Mauro Silva Sandy, and Elder José Reioli Cirilo. Estudo e avaliação de Linguagens de Programação Musical. In *Proceedings of the VIII Workshop on Ubiquitous Music (UBIMUS)*, volume 8, pages 86–97, São João del-Rei - MG - Brazil, 2018.

[27] Rodrigo Ramos de Araújo, José Mauro da Silva Sandy, Elder José Reioli Cirilo, and Flávio Luiz Schiavoni. Análise e classificação de linguagens de programação musical. *Revista Vórtex*, 6:1–24, 2018.

[28] João Teixera Araújo, Flávio Luiz Schiavoni, and Fábio dos Passos Carvalho. Mediação tecnológica e ubiquidade na participação do público em espetáculos de arte sonora. In *Anais do Poderes do som: I^a conferência internacional de pesquisa em sonoridades (Powers of sound: 1st international conference on sonorities)*, pages 148–150, Florianópolis, 2019.

[29] DE BRUNDTLAND RELATÓRIO. Nosso futuro comum. *Comissão Mundial sobre Meio Ambiente e Desenvolvimento*, 2, 2016.

[30] Marcela Alves Almeida and Flávio Luiz Schiavoni. Aspectos da sustentabilidade e colaboraÇÃo na arte digital. *Art & Sensorium*, 5:01–14, 2018.

[31] Marcela Alves Almeida and Flávio Luiz Schiavoni. Do código à colaboração: sustentabilidade na arte digital. In *Simpósio Internacional em Artes, Urbanidades e Sustentabilidade*, pages 156–164, São João del-Rei - MG - Brazil, 2017.

[32] Flávio Luiz Schiavoni. Software livre e sustentabilidade. In *Simpósio Internacional em Artes, Urbanidades e Sustentabilidade*, pages 175–184, São João del-Rei - MG - Brazil, 2017.

[33] Fábio dos Passos Carvalho and Flávio Luiz Schiavoni. Paisagens sonoras: o toque dos sinos na era digital. In *Anais da 2^a Semana Acadêmica Integrada dos Cursos de Letras / 15^a Semana Acadêmica do Curso de Letras*, pages 1–1, São João del-Rei, 2019.

[34] Fábio dos Passos Carvalho, João Teixera Araújo, and Flávio Luiz Schiavoni. Os sinos na era da ubiquidade: a relação entre as tecnologias e o patrimônio histórico. In *Anais do Poderes do som: I^a conferência internacional de pesquisa em sonoridades (Powers of sound: 1st international conference on sonorities)*, pages 153–155, Florianópolis, 2019.

[35] Thiago de Andrade Morandi and Flávio Luiz Schiavoni. Os registros ubíquos das paisagens: da imagem ao som. In *Proceedings of the VIII Workshop on Ubiquitous Music (UBIMUS)*, volume 8, pages 66–75, São João del-Rei - MG - Brazil, 2018.

[36] Flávio Luiz Schiavoni, José Mauro da Silva Sandy, Thiago Tadeu Souto Cardoso, André Lucas Nascimento Gomes, and Frederico Ribeiro Resende. O ambiente de programação visual mosaicode. In *Anais da 9^a Sessão de Ferramentas do CBSoft*, volume 1, pages 25–35. Sociedade Brasileira de Computação, 2018.

[37] Flávio Luiz Schiavoni and Luan Luiz Gonçalves. From Virtual Reality to Digital Arts with Mosaicode. In *2017 19th Symposium on Virtual and Augmented Reality (SVR)*, pages 200–206, Curitiba - PR - Brazil, Nov 2017.

[38] Flávio Luiz Schiavoni, Thiago Tadeu Souto Cardoso, André Lucas Nascimento Gomes, Frederico Ribeiro Resende, and José Mauro Silva Sandy. Utilização do Ambiente Mosaicode como ferramenta de apoio para o ensino de Computação Musical. In *Proceedings of the VIII Workshop on Ubiquitous Music (UBIMUS)*, volume 8, pages 25–32, São João del-Rei - MG - Brazil, 2018.

[39] André Lucas Nascimento Gomes, Frederico Ribeiro Resende, and Flávio Luiz Schiavoni. Desenvolvimento de extensões de processamento e síntese de imagens para a ferramenta Mosaicode. In *Proceedings of the CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES, 31 (SIBGRAPI)*, pages 1–4, Foz do Iguaçu - PR - Brazil, 2018.

[40] Flávio Luiz Schiavoni and Luan Luiz Gonçalves. Programação musical para a web com o Mosaicode. In *Anais do XXVII Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, pages 1–6, Campinas - SP - Brazil, 2017.

[41] Flávio Luiz Schiavoni, Luan Luiz Gonçalves, and André Lucas Nascimento Gomes. Web audio application development with mosaicode. In *Proceedings of the 16th Brazilian Symposium on Computer Music*, pages 107–114, São Paulo - SP - Brazil, 2017.

[42] Gabriel Lopes Rocha and Flávio Luiz Schiavoni. Gesto, linguagem e mapeamento em instrumentos musicais digitais. In *Anais da 2^a Semana Acadêmica Integrada dos Cursos de Letras / 15^a Semana Acadêmica do Curso de Letras*, pages 1–1, São João del-Rei, 2019.

A retrospective of the research on musical expression conducted at CEGeME

Mauricio A. Loureiro[1], Tairone N. Magalhães[1]*, Davi A. Mota[1], Thiago A. M. Campolina[1], Aluizio B. O. Neto[1]

[1]CEGeME - Center for Research on Musical Gesture and Expression – UFMG
Av. Antônio Carlos, 6627 – 31270-010, Belo Horizonte, MG

{mauricioloureiro, tairone, davimota, thcampolina, aluizioneto}@ufmg.br

Abstract. *CEGeME - Center for Research on Musical Gesture and Expression is affiliated to the Graduate Program in Music of the Universidade Federal de Minas Gerais (UFMG), hosted by the School of Music, Belo Horizonte, Brazil, since 2008. Focused on the empirical investigation of music performance, research at CEGeME departs from musical content information extracted from audio signals and three-dimensional spatial position of musicians, recorded during a music performance. Our laboratories are properly equipped for the acquisition of such data. Aiming at establishing a musicological approach to different aspects of musical expressiveness, we investigate causal relations between the expressive intention of musicians and the way they manipulate the acoustic material and how they move while playing a piece of music. The methodology seeks support on knowledge such as computational modeling, statistical analysis, and digital signal processing, which adds to traditional musicology skills. The group has attracted study postulants from different specialties, such as Computer Science, Engineering, Physics, Phonoaudiology and Music Therapy, as well as collaborations from professional musicians instigated by specific inquiries on the performance on their instruments. This paper presents a brief retrospective of the different research projects conducted at CEGeME.*

1 Introduction

Aiming at the empirical research on music performance, the Center for Research on Musical Gesture and Expression (CEGeME) was established in 2008, affiliated to the Graduate Program in Music of the Universidade Federal de Minas Gerais (UFMG), directed by Mauricio Alves Loureiro. Hosted by the School of Music of UFMG, at Belo Horizonte, Minas Gerais, Brazil, our laboratories have adequate facilities and equipment for the acquisition and analysis of acoustic data and body movement of musicians, which have been sponsored by grants from Brazilian research funding agencies: CNPq - National Council of Scientific and Technological Development; FAPEMIG - Foundation for Research of the State of Minas Gerais; and CAPES - Foundation for Brazilian postgraduate education. Research at CEGeME departs from musical content information retrieved from performed music, extracted from the audio signal and the three-dimensional spatial position of the musician, recorded during the performance. Analytical models proposed by CEGeME seek to establish a musicological approach to different aspects of musical expressiveness by investigating causal relations between the expressive intention of musicians and the way they manipulate the acoustic material, along with how they move while playing a piece of music. The methodology seeks support on knowledge such as computational modeling, statistical analysis and digital signal processing, that adds to traditional musicology skills.

*Supported by CAPES.

2 Research

In 2004, just a few years before the establishment of CEGeME, some of our members conducted research in musical timbre at CEFALA (Center for Research on Speech, Acoustics, Language and Music), a research group located at the School of Engineering of UFMG. They proposed a computational model for representing dynamic timbre variations on the clarinet, using a database of notes performed in four different intensities (*pianissimo* to *fortissimo*), covering the whole extension of the instrument [1, 2, 3]. The compelling results from those studies motivated Mauricio Loureiro to proceed in applying computational tools to other problems, and led him to found an interdisciplinary group focused on empirical research on music performance. The group has attracted study postulants from different specialties, such as Computer Science, Engineering, Physics and Phonoaudiology, as well as collaborations from professional musicians instigated by specific inquiries on the performance on their instruments. Over the years, a number of different research projects were conducted at CEGeME. This section presents the CEGeME's projects, which fall into four main categories: (1) Multimodal music information extraction; (2) Empirical study of performed music; (3) Acoustical component of ensemble performance; (4) Sonological applications to music learning and human development.

2.1 Multimodal music information extraction

Consists in the development and continuous improvement of tools, algorithms and computational models for extracting acoustic and kinematic parameters from both the audio signal and the three-dimensional spatial position of musicians, to support the requirements of internal research projects.

2.1.1 Models for extraction and parameterization of musical expressiveness content

The first research project developed by the group was concerned with the detection of note onsets and the extraction of acoustic descriptors from recordings of clarinet performances [4, 5, 6]. This project culminated in a Matlab tool called *Expan* [7], aimed at extracting variations in note duration, articulation, intensity, pitch, and timbre, among other features, from clarinet recordings. This tool was the outcome of several studies focused on the development of methods for segmenting musical audio signals. It has been essential for proposing and estimating descriptors for the audio parameterization required by the analyses carried by the group. Studies have focused on DSP techniques for the detection of time instants, such as note onsets, note offsets, end of attack, and beginning of release. Methods of automatic segmentation have always been evaluated by comparing output data with manual segmentation. At a certain point, the group developed a prototype graphical user interface to use the tool. Later, we decided instead, to implement a Vamp plugin for Sonic Visualizer/Audacity, containing the algorithms and methods of Expan, capable to provide a straightforward way for comparing parameter values and visualizing data in an interactive graphical interface.

The group is currently working on the development of a Python library called *Iracema*, which proposes an architecture containing abstractions for the easy manipulation of time series and extraction of information from audio. This tool introduces new methods and techniques, moving towards machine learning models, in order to improve onset detection and obtain more meaningful musical information from audio [8, 9].

2.1.2 Content analysis of note attack and note transitions in music performance

Characteristics related to the quality of the transitions between consecutive notes are decisive in the construction of an interpretation. They are manipulated by the performer by controlling note durations and the quality of attacks and note groupings. Studies on this project focus on the modelling of what happens in the transient region when notes are initiated, as well as in the time interval situated between consecutive notes of a musical sentence [10, 11, 12, 13]. Description of note attack transients also explored the harmonic content of reverberation independently from the note being played, which allows inferences about the musician's interaction with room acoustics [14, 15].

2.1.3 Segmentation and parameterization of musician's physical gesture

Studies concerned on the extraction of information from musician's body movement, while playing their instruments. Three-dimensional spatial position of their body and instruments are tracked by two high-end 3D motion capture devices: a more accurate with active infrared markers (Optotrak Certus), and a more versatile with passive infrared markers (Optitrack Motive/Flex3). Techniques were developed for processing and segmenting movement data, based on statistical, geometrical, temporal and musical aspects, aiming at developing methodologies for multimodal parametric analysis of music performance, integrating musician's physical gestures information to acoustical parameters extracted from audio. [16, 17, 18, 19, 20].

2.2 Empirical study of performed music

Focusing on different aspects of musical expressiveness, these projects propose analytical models that seek to identify causal relations between the expressive intention of musicians and the way in which they manipulate the acoustic material and how they move when interpreting a musical work.

2.2.1 Individuality, consistency and diversity of music performances

Studies in the realm of this project focus on analyzing the dynamics of acoustic and gestural parameters observed during music performances, aiming to characterize traits related to individual style, i.e., common aspects shared by different performances played by the same musician [21]. These studies can be used to support models for individuals discrimination (the automatic recognition of performers) as well as for characterization of performance style [22, 23, 24]. Considering that temporal variations are the acoustic manipulations that most stand out in the perception of the expressive quality of an interpretation of a score of traditional classical repertoire, some studies focused on the fluctuation of temporal proportions of rhythmic structures produced by the interpreter, to investigate its potential for characterizing such expressive manifestations [25, 26]. As the relevance of expressive manipulations may shift to other types of acoustic parameters for the characterization of individuality, depending on factors such as genre and cultural context, another study focused on pitch manipulations of bendings and vibratos on the electric guitar. The results showed a remarkable consistency within performances by the same guitarist, but still with substantial diversity when comparing different performers. [24, 27].

2.2.2 Body movement in music performance: role and meaning

This project comprises studies focused on the hypothesis that body movements executed by musicians during performances are closely related to their sounded expressive intentions according to their understanding of the musical structure they interpret. Sophisticated recurrence analysis of body movement patterns, identified in different performances of the same music, revealed that musicians execute gestures patterns recurrently associated to higher expressive content dictated by outstanding elements of the musical structure, such as melodic phrasing, score dynamic markings, rhythmic and harmonic transitions [28, 29, 30, 31]. Studies in the realm of this project also showed evidences that musician's gestures reflect musical phrase organization [32], as well as that body movement during music performances carries enough information for recognizing the interpreter [23].

2.3 Interaction and coordination in ensemble music performance

Aimed at investigating the synchronization/coordination process in small musical ensembles, projects in this category attempt to model musician's interactions in typical performance scenarios, using visual and acoustical responses collected during real musical performances simulations. Patterns of temporal synchronization and/or other coupling patterns, manifested through sound manipulations and body movements, were observed and analyzed, in order to map acoustic and gesture-related elements that could influence the adjustment between musicians.

2.3.1 Acoustical component of ensemble performance

This topic addresses the temporal synchronization between musicians, the dynamics of the interaction between leaders and followers and how it may influence the coordination of ensembles. In ensemble performances, musicians share the responsibility of shaping the sounds, either by serving as a reference for other players (e.g. a conductor) or by following the cues indicated by the ensemble leader. The dynamics of this interaction seems to extends beyond the temporal synchronization, encompassing multiple dimensions of the musical domain, for instance, as we observed that flute players tend to modify their timbre when following a clarinet or a bassoon with consistently different tendencies for each instrument [33, 34, 32]. Another central question of these studies is how musicians can become effective in performing such coordinated actions, either when they play as leaders or as followers. According to a known hypothesis of cognitive psychology, the act of anticipating the effects of other peoples' actions may be in the center of the coordination process. One consequence of this hypothesis is that if internal models formed in previously performed actions are applicable to current actions, musicians should be better in predicting the results of their own recorded actions than the results of actions performed by others [35]. Our studies corroborate this hypothesis by demonstrating that musicians have better chance of synchronizing with their own previously recorded performances [36, 37, 38].

2.3.2 Gestural component of ensemble performance

The focus is on the gestural interactions between musicians during ensemble performance. Recurrent patterns of body/instrument movements are identified in an attempt to map acoustical responses to gesture-related elements that could influence the adjustment between musicians. While verifying the relationship of musician's body movements to their expressive intentions during the performance, we were able to identify information patterns recurrent enough for recognizing who played the excerpt, which we call "gestural signatures". We also observed that these signatures were disturbed, when musicians followed recordings. Moreover, disturbances were smaller when musicians followed recordings made by themselves, even without visual contact between co-performers [39, 40].

2.3.3 Multimodal interactions in ensemble performance

Studies regarding the inherent multimodal complexity in music ensemble interactions, which uses complementary approaches involving the analysis of kinematic and acoustic parameters as a way to model human-to-human musical interactions by combining the multiple dimensions that are enrolled in the process of creating collaborative musical performances [41, 37, 42].

2.4 Sonological applications to music learning and human development

It takes as object of investigation the human behavior in musical activities mediated by technological tools directed to learning and human development.

2.4.1 Multimodal data analysis in musical expertise development

Several tools for analyzing multimodal data extracted from music performance are currently being developed and advancements in this area evolve in an astonishing pace. This project investigates how such techniques can be used on the refinement of the expertise and skill development by students, teachers and professional musicians. A Master Thesis investigated left hand position shifting technique on the double bass, relating kinematic and acoustic information [43]. Timbre descriptors were used to verify timbre modification due to coupling interpretation in ensemble performance of different instruments [33]. An attempt was made to use the outcome of this study on the construction of a flute-bassoon duo performance [34]. The potentials indicated by these studies motivated further investigation on the interaction of musical practice in a classroom setting with audio analysis tools [44]. A PhD Thesis reported methods and results for this interaction, occurred on Weekly clarinet classes, in which the master and his class were recorded and analyzed, as extracted musical information were used as feedback for both teacher and students [45]. Another thesis conducted by an accomplished oboist focused the relationship of oboe reed scraping and sound quality and sound emission techniques. In 2015 an harpist joined CEGeME willing to investigate what role musical gesture plays on helping the performer to organize the expressive ideas behind his/her playing. She began her work by inquiring how feedback can aid the development of metacognitive skills for music performance [46, 44] and later delved into understanding how this process occurs with undergraduate level students [47] and [48]. A professional clarinetist, member of one of the most prominent Brazilian symphony orchestras is currently delving into the relationship between clarinetists' gestures and the quality of the *legato*. He conducts computational analysis of sound and fingers' movement acquired from a series of experiments with a number of professional musicians, using motion capture devices and high quality audio recording. [49, 50]. Two master dissertations are on the first stages of development within this project. One deals with didactic approaches regarding the use of valves on the bass trombone [51], and the other investigates the relationship between airflow speed and sound quality in trumpet playing [52].

2.4.2 Sonological Applications to Human Development

The first half of the twentieth century gave birth to a sonic revolution, consequence of the invention of devices capable of recording and playing back sounds, allowing for novel ways of investigating how humans respond to and act through sound. Within the domains of this project we seek to understand human conditions and practices through its relation to sound and sonology-related themes in general. Most of the work presented here is related to health and well-being, where music-therapy plays a prominent and increasingly important role. A Master project sought to adapt the Nordoff Robbins Musical Communicativeness Scale to the Brazilian context, evaluating cultural and linguistic issues related to its use. The scale is used in the USA since the 1960's to evaluate behaviors from sound and musical stimuli through vocalizations, manipulation of musical instruments and body movement. The analysis of 24 music therapy methodological videos was adopted for boardline cases' people with neurodevelopmental disorders. One researcher and four invited examiners have participated in this study stage. The inter-examiner scores have presented moderate and strong correlations (Spearman), indicating evidences of reliability for the Brazilian version of the scale. [53, 54, 55, 56]. The dissertation led to an ongoing PhD project [57], in which the relationship between Pierre Schaeffer's modes of listening and the Nordoff Robbins Scales is being studied[58]. A musical game was proposed in a Master project in order to investigate expressive behaviour in children suffering from disorders within the autism spectrum, which proposed 12 principles and strategies and 7 recommendations for games researchers [59, 60, 61]. Two master projects are being developed, on behavioral evaluation of premature babies in response to sound stimuli [62] and on how music can help with arm abduction in people with cerebral palsy.' [63].

3 Workshops and scientific events

3.1 Human movement and music: capture, representation, analysis

The seminar was held at the School of Music of UFMG, on June 7th and 8th, 2017. Invited researchers from local and international institutions, as well as collaborators of CEGeME presented lectures, recitals, and conducted seminars and workshops [64]:

- Carolina Brum Medeiros from the Input Devices and Music Interaction Laboratory (IDMIL), School of Music, McGill University, Montreal, conducted a workshop, *"Motion Capture - solutions for specific issues at a motion capture laboratory"*, and presented the lecture *"Processing and analysis of human movement - methodological strategies for the processing and analysis of human movement"*.
- Hani Camille Yehia, director of the Center for Research on Speech, Acoustics, Language and Music (CEFALA), School of Engineering, Federal University of Minas Gerais (UFMG) presented the lecture *"Comparative Analysis of Techniques of Acquisition and Representation of the Human Movement"*.
- Luiz Bavaresco de Naveda, School of Music, State University of Minas Gerais (UEMG) conducted a workshop, *"Analysis of cross-modal patterns through estimation of directional changes density applied to musical movement"*.
- André Cavazotti e Silva, School of Music, Federal University of Minas Gerais (UFMG) presented a report about the work in progress for a performance involving the body movement of a violinist with video capture, *"The flight of Tchiurubibirú in search of the first movements"*.

The seminar ended with a discussion panel on the research projects conducted at CEGeME involving the invited researchers, follwed by a harp recital performed by Aricia Marques Ferigato, a PhD student at CEGeME.

3.2 Musical expressiveness descriptors: meaning and applications

Inspired by the festival "Pint of Science", this event consisted in a lecture held at Restaurante Novo Paladar, in Belo Horizonte, across the street of UFMG Campus, on June 21st, 2017 [65], presented by Jônatas Manzzolli, former irector of the Interdisciplinary Nucleus of Sounding Communication (NICS), Institute of Arts, State University of Campinas (UNICAMP) and visiting professor at the Center for Autonomous Systems and Neuro-robotics (NRAS), Pompeu Fabra University, Barcelona, Spain.

Manzzolli presented a research project on interfaces and devices in an immersive laboratory for generating information from motion, image and sound with potentials strategic applications for the study of musical expressiveness, in its interpretative and creative aspects. Iannis Xenakis's on music motivated the debate that followed the lecture: "I do not think that any attempt to consider music as a language can be successful. The substructure of music is much closer to the substructure of space and time. Music is purer and much closer to the categories of the mind."

3.3 Technical and learning aspects of clarinet and bass clarinet: sonority, articulation, extended technique

Held at the School of Music of UFMG, on October 2017 [66], the seminar had as special guest, the renowned Brazilian clarinetist Luis Afonso Montanha, of the Music Department of the University of São Paulo (USP), former first clarinetist of the Municipal Symphony Orchestra of São Paulo and the Symphonic Jazz Orchestra of the State of São Paulo, winner of several national and international Music Prize.

- Montanha, together with the permanent clarinetists of Philharmonic Orchestra of Minas Gerais, Alexandre Silva, Ney Franco and Marcus Julius Lander, conducted a discussion panel on sound emission (timbre and tuning) and articulation (attack, legato, staccato) on the clarinet.
- These same artists participated on the Seminar *Processes of understanding and awareness on clarinet and bass clarinet learning*, coordinated by Aluizio Barbosa de Oliveira Neto, a research member of CEGeME, PhD

candidate at the time. The seminar proposed experiments on empirical analysis of clarinet performances, aimed at discussing possible contribution of parametric feedback to the development of skills related to aspects of sound emission, such as sound quality and tuning, as well as of the execution of different types of note articulation, such as *legato* and *staccato*.

- Montanha also gave a clarinet and bass clarinet Masterclass with emphasis on contemporary repertoire and extended techniques and presented the lecture, *Extended technique on bass clarinet and the relation between composer and performer in the development of a musical work.*

3.4 11th International Conference of Students of Systematic Musicology (SysMus)

On June 2018, CEGeME hosted the 11th International Conference of Students of Systematic Musicology (SysMus), which happened for the first time in Latin America, in the city of Belo Horizonte, and received 60 submissions from from 47 universities and institutes around the world [67]. The event was attended by more than 60 participants, from 10 different countries, and included three speeches by prominent researchers in the field.

- Juan Pablo Bello, associate professor of Music Technology and Computer Science & Engineering at New York University – *"Some Thoughts on the How, What and Why of Music Informatics Research"*
- Andre Holzapfel, assistant professor at the Media Technology and Interaction Design department at KTH Royal Institute of Technology in Stockholm, working in the area of Sound and Music Computing (SMC) – *"Computer-aided studies of rhythm and meter and the impact of ethnocentric bias"*
- Mauricio Alves Loureiro, director of CEGeME and full professor of music at the Federal University of Minas Gerais (UFMG) – *"Systematic Musicology: a perspective from the musical signal"*

4 Artistic events

In 2015, some of the members of our group gave birth to the NA/CEGeME, an artistic group aiming to put into practice the technical knowledge generated by the laboratory activities through art. Three performances were developed: *ApropriaAção* [68], *Espaço, Fluxo e Transcendência* [69] and *Subverta* [70]. The pieces were intended to arouse different sensibilities, transforming sites located in a grove at the university campus into an artistic space.

4.1 ApropriAção

ApropriAção was intended to transform places that until then carried out various functions into artistic spaces, presenting performances that involve different aspects of human perception. Visual projections were used as a way to contribute to the sonic environment created by the musicians during the performance, producing sounds and images computationally modulated by the movements and responses from the audience.

4.2 Espaço, Fluxo e Transcendência

In this performance, the woods in front of the School of Music of UFMG were transformed into a space of sound and visual interaction. The public was encouraged to traverse paths where, through their own movement, they were able to manipulate the acoustic material generated by the various performances taking place on the audience surroundings. Immersed in sound and guided by it, the flow of people also controlled the visual projections that illuminate the possible paths. Sounds and images resulting from the combined action between artists and public emerge from the woods and are responsible for the resulting experienced environment.

4.3 SubvertA!

In a happening entitled *SubvertA!*, CEGeME's Artistic Group entices the audience to engage on novel experiences with their surroundings. Through aesthetic fruition and by promoting the creative use of public spaces, NA/CEGeME developed performances involving different aspects of human perception. Threaded by music, sound, visual projections and multisensory interactions, SubvertA! promotes a challenge to the experience of university spaces, seeking to strengthen resistance to actions that inhibit and discourage the creative use of the campus, converging with other sociopolitical and cultural manifestations and occupation policies in the urban context. Audio processing tools and techniques for motion analysis developed at the CEGeME Lab were used to create the audio/visual landscape and interactive devices that allowed for this experience.

Figure 1: NA/CEGeME performing in 2017.

5 Future perspectives

Results of CEGeME research show potentials to broaden the perception of musical expressiveness content, which could be directed to the daily musical practice of instrumentalists and singers, that could contribute to the development of performance excellence, as well as to be used as pedagogical resource in musical instrument and singing lessons. From this perspective, we envisage in CEGeME the development of an appropriate platform that can offer professional musicians, teachers and instrumentalist students an interactive interface that demands low training investment from the user in order to provide a simpler and more comfortable tool for the empirical research of musical performance, aiming at facilitating the use by a greater number of users and for a greater volume of data. We believe that such a tool, properly tested and validated in a real situation of practice and musical pedagogy, should present great potential of application in different contexts of the practice and the teaching of music. The results of this project and possible resources produced by it may also arouse interest in research in other areas of music, such as interactive composition, music education and music therapy, as well as other arts such as theater, dance and art education.

6 Acknowledgements

CEGeME projects are made possible with the sponsorship of the National Council for Scientific and Technological Development - CNPq and the Foundation for Research of the State of Minas Gerais - FAPEMIG.

References

[1] Hugo B. Paula, Hani C. Yehia, J. A. Vasconcelos, and Mauricio A. Loureiro. Representing timbre dynamics of a musical instrument: Comparison between GA and PCA. In *Machine Learning for Signal Processing XIV - Proceedings of the 2004 IEEE Signal Processing Society Workshop*, 2004.

[2] Mauricio A. Loureiro, Hugo B. Paula, and Tairone N. Magalhães. Modelagem da variação do timbre musical utilizando modelos auditivos e mapas de Kohonen. In *I Encontro Nacional de Cognição e Artes Musicais*, pages 156–163, Curitiba, 2006.

[3] Mauricio A. Loureiro and Hugo B. Paula. Timbre de um instrumento musical: caracterização e representação. *Per Musi – Revista Acadêmica de Música*, 14:57–81, 2006.

[4] Mauricio. A. Loureiro, Tairone N. Magalhães, Rodrigo C. Borges, Thiago A. M. Campolina, Davi A. Mota, and Hugo de Paula. Segmentação e extração de descritores de expressividade em sinais musicais monofônicos. In *III Seminário Música Ciência e Tecnologia*, pages 109–125, 2008.

[5] Mauricio. A. Loureiro, Rodrigo C. Borges, Thiago A. M. Campolina, Tairone N. Magalhães, Davi A. Mota, and Hugo De Paula. Extração de conteúdo musical em sinais de áudio para análise de expressividade. In *Anais do XVII Encontro da Sociedade Brasileira de Acústica SOBRAC*, pages 146–152, Belo Horizonte, 2008.

[6] Mauricio A. Loureiro, Tairone N. Magalhães, Rodrigo C. Borges, Thiago A. M. Campolina, Davi A. Mota, and Hugo B. de Paula. Segmentação automática de sinais musicais monofônicos para análise da expressividade. In *Congresso da Associação Nacional de Pesquisa e Pós-Graduação (ANPPOM)*, pages 334–340, 2008.

[7] Thiago A. M. Campolina, Davi A. Mota, and Mauricio A. Loureiro. Expan: a tool for musical expressiveness analysis. In *Proceedings of the 2 nd International Conference of Students of Systematic Musicology*, pages 24–27, 2009.

[8] Tairone N. Magalhães. *Computational Tools and Methods for the Empirical Study of Monophonic Music Performance*. Ongoing PhD Thesis, Universidade Federal de Minas Gerais, 2019.

[9] Willy G. Cornelissen. Onset detection in silico. Ongoing Master Thesis, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, 2019.

[10] Mauricio A. Loureiro, Hani C. Yehia, Hugo B. Paula, Thiago A. M. Campolina, and Davi A. Mota. Content analysis of note transitions in music performance. In *Proceedings of the 6th Sound and Music Computing Conference, SMC 2009*, pages 355–359, 2009.

[11] Mauricio. A. Loureiro, Davi A. Mota, Thiago A. M. Campolina, Euler C. F. Teixeira, Hugo Bastos De Paula, and Hugo B. Paula. O Som entre as Notas: Representação das Transições entre as Notas em Instrumentos Monofônicos. In *Anais do XIX Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, pages 581–585, 2009.

[12] Thiago A. M. Campolina. Estudo sobre Transientes em Sinais de Fala e Música. Master thesis, Universidade Federal de Minas Gerais, 2012.

[13] Rodrigo C. Borges. Um Estudo sobre Classificações Físicas e Perceptivas de Timbres da Escuta Cotidiana, de Sons Sintetizados e de Noise Music. Master dissertation, Universidade Federal de Minas Gerais, 2013.

[14] Thiago A. M. Campolina and Mauricio A Loureriro. Description of Monophonic Attacks in Reverberant Environments via Spectral Modeling. In *Paper submitted to Computer Music Multicisciplinary Research Conference, Marseille, France, 2019.*, Marseille, France, 2019.

[15] Campolina, Thiago A. M. *Detecção, análise espectral e caracterização de ataques em instrumentos monofônicos*. Ongoing PhD Thesis, Universidade Federal de Minas Gerais, 2019.

[16] Hugo B. Paula, Euler C. F. Teixeira, Raphael A. V. Borges, and Mauricio A. Loureiro. Automatic Gesture Segmentation of a Music Performance Using a Cumulative Dissimilarity Measure. In *Brazilian Symposium on Computer Music*, number 2005, pages 13–24, 2015.

[17] Euler C. F. Teixeira. Análise Quantitativas da Expressividade Musical com Base em Medidas Acústicas e do Gesto Físico. Master dissertation, Universidade Federal de Minas Gerais, 2010.

[18] Euler C. F. Teixeira, Hani C. Yehia, and Mauricio A. Loureiro. Aspectos metodológicos da Pesquisa em Expressividade Musical a partir de Informação de Movimento Corporal. In *Anais do XX Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, pages 1364–1370, 2010.

[19] Euler C. F. Teixeira, Hani Camille Yehia, Mauricio A Lourerіro, and Marcelo Wanderley. Motion recurrence analysis in music performances. In *SMC Proceedings of Sound and Music compouting Conference 2013*, number August, pages 317–322, Stockholm, Sweden, 2013. KTH.

[20] Euler C. F. Teixeira. *Motion analysis of clarinet performers*. PhD thesis, Universidade Federal de Minas Gerais, 2014.

[21] Tairone N. Magalhães, Davi A. Mota, and Mauricio A. Loureiro. A individualidade musical na performance: conceitos teóricos, estratégias metodológicas e perspectivas. In *Anais do 13º Simpósio de Cognição e Artes Musicais - SIMCAM 13*, pages 270–277, Curitiba, Brazil, 2017.

[22] Gabriel P. Pezzini, Thiago A. M. Campolina, and Mauricio. A. Loureiro. Stochastic Modeling of Expressiveness: Representing the Temporal Evolution of the Descriptors Using HMM Modeling. In *SysMus14 – International Conference of Students of Systematic Musicology*, pages 1–4, 2014.

[23] Davi A. Mota, Mauricio A. Loureiro, and Rafael Laboissière. Identificação do intérprete a partir de

informação de movimento corporal. *Anais do XIX Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, 2014.

[24] Tairone N. Magalhães. Análise do bending e do vibrato na guitarra elétrica a partir dos descritores de expressividade da ferramenta Expan. Master thesis, Universidade Federal de Minas Gerais, 2015.

[25] Mauricio A. Loureiro and Davi A. Mota. Caracterização Paramétrica da Agógica na Interepretação Musical. In *XXVIII Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música – Manaus - 2018*, pages 1–10. ANPPOM - Associação Nacional de Pesquisa e Pós-Graduação em Música, 2018.

[26] Mauricio A. Loureiro. Uniformidade e Diversidade em Execução Musical: mesa redonda da ANPPOM discute a agógica. In Sonia Regina Albano de Lima and Martha Tupinambá de Ulhôa, editors, *ANPPOM: 30 Anos de Criação*, pages 72–89. ANPPOM - UDESC, Florianópolis, 2018.

[27] Tairone N. Magalhães, Davi A. Mota, Aluizio B. O. Neto, Mauricio A. Loureiro. Análise do vibrato e bending na guitarra elétrica. In *Anais do XV Simpósio Brasileiro de Computação Musical*, pages 36–47, 2015.

[28] Euler C. F. Teixeira, Hani C. Yehia, and Mauricio A. Loureiro. Relating movement recurrence and expressive timing patterns in music performances. *Journal of the Acoustical Society of America*, 138(3), 2015.

[29] Euler C. F. Teixeira, Mauricio A. Loureiro, Marcelo M. Wanderley, and Hani C. Yehia. Motion Analysis of Clarinet Performers. *Journal of New Music Research*, 44(2):97–111, 2015.

[30] Euler C. F. Teixeira and and Hani C. Yehia Mauricio A. Loureiro. Linking movement recurrence to expressive patterns in music performance. In Micheline Lesaffre, Marc Leman, and Pieter-Jan Maes, editors, *The Routledge Companion to Embodied Music Interaction*, pages 380–387. Routledge Taylor & Francis Group, New York, 2017.

[31] Euler C. F. Teixeira, Mauricio A. Loureiro, and Hani C. Yehia. Expressiveness in Music From a Multimodal Perspective. *Music Perception: An Interdisciplinary Journal*, 36(2):201–216, dec 2018.

[32] Thaís F. Santos. *The relationship between ancillary gestures and musical phrase organization: application to flute performance*. Phd thesis, Universidade Federal de Minas Gerais, 2017.

[33] Thais F. Santos, Aluizio B. O. Neto, and Mauricio A. Loureiro. Musical communication of flutists in ensemble performance. In *7th International Conference of Students of Systematic Musicology (SysMus14)*, pages 1–5, London, 2014.

[34] Thaís F. Santos, Raquel S. Carneiro, Aluizio B. O. Neto, and Mauricio A. Loureiro. Construção de uma performance em conjunto a partir de um estudo de parâmetros acústicos. In *Associação Nacional de Pesquisa e Pós-Graduação em Música*, 2015.

[35] G. Knoblich and R. Flach. Predicting the effects of actions: Interactions of perception and action. *Psychological Science*, 12(6):467–472, 2001.

[36] Rafael Laboissière, Davi A. Mota, Thiago A. M. Campolina, Hani C Yehia, and Mauricio A. Loureiro. Evidence for Self-Other Effects and Structure-Related Attentional Mechanisms in the Primo/Secondo Performances of Clarinetists. In UFMG, editor, *Proceedings of the 16th International Conference on Perception and Action (ICPA 2011)*, Ouro Preto, Minas Gerais, Brazil, 2011.

[37] Davi A. Mota. Análise dos padrões de sincronização em duos de clarineta a partir de parâmetros acústicos e cinemáticos. Master dissertation, Universidade Federal de Minas Gerais, 2012.

[38] Mauricio A. Loureiro, Davi A. Mota, Thiago A. M. Campolina, Hani C. Yehia, and Rafael Labiossière. Acoplamento interpretativo e sincronização em duos de clarinetas. *OPUS*, 18(2):181–206, 2012.

[39] Davi A. Mota, Mauricio A. Loureiro, and Rafael Laboissière. Influence of expressive coupling in ensemble performance on musicians' body movement. In *Proceedings - 40th International Computer Music Conference, ICMC 2014 and 11th Sound and Music Computing Conference, SMC 2014 - Music Technology Meets Philosophy: From Digital Echos to Virtual Ethos*, number September, pages 513–519, Athens, Greece, 2014. The National and Kapodistrian University of Athens.

[40] Davi A. Mota, Mauricio A. Loureiro, and Rafael Laboissière. *Gestural interactions in ensemble performance*. Routledge Taylor & Francis Group, New York, 2017.

[41] Davi A. Mota, Mauricio A. Loureiro, and Rafael Laboissière. Analysis of synchronization patterns in clarinet duos using acoustic and kinematic parameters. In Favio Shifres, María Jacquier, Daniel Gonnet, María Inés Burcet, and Romina Herrera, editors, *Actas de ECCoM. Vol. 1 No1, "Nuestro Cuerpo en Nuestra Música. 11o ECCoM"*, volume 1, pages 199–206, Buenos Aires, 2013. SACCoM.

[42] Davi A. Mota. *Multi-modal coupling in musical performance*. Phd thesis, Universidade Federal de Minas Gerais, 2017.

[43] Giordano C. O. M. Cornelio. Análise acústica e cinemática da técnica de mudança de posição no contrabaixo acústico. Master's thesis, 2013.

[44] Aluizio B. O. Neto, Arícia Ferigato, Thiago A. M. Campolina, Ravi S. V. Domingues, and Mauricio A. Loureiro. Ferramentas de análise empírica e metacognição no processo de interpretação musical. *Anais do XII Simpósio Internacional de Cognição e Artes Musicais*, pages 250–257, 2016.

[45] Aluizio B. O. Neto. *Rumo à sala de estudos aumentada: Experiências com suporte computacional para o desenvolvimento técnico e perceptivo na performance musical*. Phd thesis, Universidade Federal de Minas Gerais, 2018.

[46] Arícia Ferigato and Mauricio A. Loureiro. Imagética musical e metáfora como recurso ativador/indutor do gesto corporal na interpretação e na performance musical. In *Anais do XIII Simpósio Internacional de Cognição e Artes Musicais*, pages 287–293, curitiba, PR, Brazil, 2017. UFPR.

[47] Arícia Ferigato and Mauricio A. Loureiro. Gesto corporal e conteúdo expressivo na construção da performance musical de alunos de harpa. In *XXVIII Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música – Manaus - 2018*, pages 1–7. ANPPOM - Associação Nacional de Pesquisa e Pós-Graduação em Música, 2018.

[48] Aricia M. Ferigato. *Construção da performance na harpa de pedais: relações entre gesto corporal e conteúdo expressivo musical*. Ongoing PhD Thesis, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, 2019.

[49] Alexandre P. Silva. *Estudo do legato na clarineta através de parâmetros acústicos e cinemáticos do movimento dos dedos*. Ongoing PhD Thesis, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, 2019.

[50] Alexandre P. da Silva. Captura dos movimentos dos dedos de clarinetistas: descrição dos processos metodológicos. In

Anais do IV Simpósio Brasileiro de Pós-Graduandos em música, pages 1095–1104, Rio de Janeiro, RJ, 2016.

[51] Renato R. Lisboa. O uso das válvulas do trombone baixo: questões técnicas do instrumento e suas consequências no repertório, performance e abordagens didáticas. Ongoing Master Thesis, Programa de Pós-Graduação em Música Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, 2019.

[52] Gustavo M. Oliveira. Influência da velocidade do fluxo de ar na sonoridade do trompete. Ongoing Master Thesis, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, 2019.

[53] Aline M. André; Cristiano M. A. Gomes; Cybelle M. V. Loureiro. Reliability inter-examiners of the nordoff robbins musical communicativeness scale brazilian version. 2018.

[54] Aline M. André; Cristiano M. A. Gomes; Cybelle M. V. Loureiro. Equivalência de itens, semântica e operacional da versão brasileira da escala nordoff robbins de comunicabilidade musical. *OPUS*, 23(2):197–215, 2017.

[55] Aline M. André, Cristiano M. A. Gomes, and Cybelle M. V. Loureiro. Escalas nordoff robbins: uma revisão bibliográfica. *Percepta-Revista de Cognição Musical*, 3(2):117, 2016.

[56] Aline M. André. Tradução e validação da Escala Nordoff Robbins de Comunicabilidade Musical. Master dissertation, 2017.

[57] Aline M. André. *Tradução e Validação das Escalas Nordoff Robbins: "Relação Criança Terapeuta na Experiência Musical Coativa" e "Musicabilidade, Formas de Atividade, Estágios e Qualidades de Engajamento"*. Ongoing PhD Thesis, Programa de Pós-Graduação em Música Universidade Federal de Minas Gerais, 2019.

[58] Aline M. André and Cybelle M. V. Loureiro. Modos da escuta de pierre schaeffer e escalas nordoff robbins: um estudo de caso. In *Congresso da Associação Nacional de Pesquisa e Pós-Graduação (ANPPOM)*, 2019.

[59] Débora L. Gomes and Cybelle Loureiro. Proposta de um aplicativo na hipersensibilidade sonora no autismo. *XIII SIMCAM-Simpósio Internacional de Cognição e Artes Musicais*, 22(2005):8, 2017.

[60] Débora L. Gomes and Cybelle Loureiro. Aplicativo musical na hipersensibilidade sonora nos transtornos do neurodesenvolvimento: Um estudo de revisão, 2017.

[61] Debora L. Gomes. Musical App in Hypersensitivity to Sounds and Neurodevelopmental Disorders: Applicable Strategies. Master Thesis, Programa de Pós-Graduação em Música Universidade Federal de Minas Gerais, 2018.

[62] Rhainara L. C. Ferreira. Validação De Um Protocolo De Avaliação Comportamental De Bebês Prematuros Em Resposta A Estímulos Sonoros. Ongoing Master Thesis, Programa de Pós-Graduação em Música Universidade Federal de Minas Gerais, 2019.

[63] Emily H. P. Ferreira. Como a música pode auxiliar na abdução de braços em pessoas com Paralisia Cerebral. Ongoing Master Thesis, Programa de Pós-Graduação em Música Universidade Federal de Minas Gerais, 2019.

[64] Thais F. Santos Aluizio B. O. Neto, Arícia M. Ferigato; Davi A. Mota and Mauricio A. Loureiro. Seminar: Human Movement and Music: capture, representation, analysis. School of Music of the UFMG, June 2017.

[65] Jônatas Manzzolli. Musical expressiveness descriptors: meaning and applications. Restaurante Novo Paladar, Av. Antônio Abrahão Caran, 764 – Pampulha, June 2017.

[66] Luis A. Montanha. Seminar: Technical and learning aspects of clarineta and bass clarinet: sonority, articulation, extended technique. School of Music of the UFMG, October 2017.

[67] SysMus18. In Tairone N. Magalhães and Davi A. Mota, editors, *Proceedings of the 11th International Conference of Students of Systematic Musicology*, 2018.

[68] NA/CEGeME Group. Performance ApropriAção. 47o Festival de Inverno da UFMG, july 2015. [www.ufmg.br/festivaldeinverno/2015/eventos; www.youtube.com/watch?v=1V0Iv3ZnyGw; accessed 31-Maio-2019].

[69] NA/CEGeME Group. Performance Espaço, Fluxo e Transcendência. 48o Festival de Inverno da UFMG, july 2016. [www.ufmg.br/festivaldeinverno/2016/eventos; accessed 31-Maio-2019].

[70] NA/CEGeME Group. Performance Subverta. 49o Festival de Inverno da UFMG, july 2017. [www.ufmg.br/festivaldeinverno/2017/eventos; www.youtube.com/watch?v=fMQUvOb52Ac; accessed 31-Maio-2019].

MusTIC: Research and Innovation Group on Music, Technology, Interactivity, and Creativity

Filipe Calegario, Giordano Cabral , Geber Ramalho

[1]MusTIC / Centro de Informática / Universidade Federal de Pernambuco
Av. Jornalista Aníbal Fernandes, s/n - Cidade Universitária (Campus Recife) - 50.740-560 - Recife - PE

fcac@cin.ufpe.br, grec@cin.ufpe.br, glr@cin.ufpe.br

Abstract. *MusTIC is a research and innovation group concerned in conceiving and developing products and experiences that have an impact on music, education, visual and performing arts, and entertainment. In particular, we have been working with tools, methods, and concepts from physical computing, interaction design, and signal processing to build new interfaces for artistic expression, to develop tools for rapid prototyping, and to improve education through robotics and gamification.*

1 Introduction

In 1997, the "Centro de Informática" at Federal University of Pernambuco, in Recife, Brazil, started to develop research activities on computer music. This effort was led by Geber Ramalho, after returning from his Ph.D. in Paris 6, under the supervision of Jean-Gabriel Ganascia and François Pachet. In 1999, a 60-hour annual course on computer music for both graduate and undergraduate students had started to be offered. During the next 13 years, this research work led to 3 Ph.D. thesis and 12 MSc dissertations.

All these previous activities on computer music set the basis for the creation of MusTIC research group in 2011 when Giordano Cabral arrived. The group, whose name is an agglutination of Music, Technology, Interactivity, and Creativity, widened the scope of the research from music to other related domains. It also incorporated other researchers and practitioners from different domains that as mechatronics, electronics, graphic expression, dance, lutherie, and others.

Since then, 4 Ph.D. thesis and 16 MSc dissertations have been presented. Beyond academic research, MusTIC has a strong orientation to innovation. Indeed, MusTIC prioritizes works that can potentially lead to concrete results for the society and works in which the technology is the means for creating new artistic, educational, or entertainment experiences. This orientation pushed the group to be involved in dozens of shows, exhibitions, or interactive installations. All this put MusTIC among the most active Brazilian research and innovation groups in art, technology, and interactivity.

2 Work Axes

Currently, the group has defined five leading research and innovation axes to guide the efforts of the students, researchers, practitioners, and professors.

2.1 Digital Musical Instruments

New artifacts are changing the way we interact with machines, and this is particularly important for the musical domain. They open a wide range of possibilities in the creation of Digital Musical Instruments (DMIs), artifacts with inputs (interface controllers) and outputs (sound synthesis modules) that connect according to a mapping strategy [1]. Contrary to acoustic instruments, which impose physical constraints on their design and fabrication, the design of DMIs has more freedom. Paradoxically, this advantage is a problem, since there is no established method or tool to guide the DMI designer or luthier. In this research line, we focus on developing prototyping tools, methods for DMI evaluation, existing technology evaluation, news sensors, and interfaces.

2.2 Sound and Music Analysis

Digital music provided the tools to a revolution in the understanding of sound and music. We are particularly interested in extracting information that even a human with a well-trained ear cannot do, or at least express. By music analysis, we mean the discovery of patterns in data represented in musical form. For example, MIDI/MusicXML files, music scores, tablatures, and MIDI/OSC events captured in live performances. We performed a wide range of researches in this field, from functional harmonic analysis to the discovery of microtiming and microdynamics patterns, from modeling rhythm and variation to the visualization of common harmonic paths by composers. By sound analysis, we mean the extraction of information directly from raw data, such as WAV/MP3 files or microphone audio streams. It involves the development of new technology to create games, the automatic detection of dangerous sounds (like gunshots and explosions) for public security, and music information retrieval (MIR) applications (such as automatic chord recognition). One of the main results of this research is our contribution to projects at the MusiGames Studio and the Audio Alerta initiative.

2.3 Automatic Accompaniment Systems

"Music improvisation and accompaniment systems" are systems that can automatically generate melodic lines, rhythmic patterns and chord voicing chaining in a particular musical style. In our research, we employ different AI techniques (e.g., Hidden Markov Models, Neural Networks, Case-based Reasoning, Multi-agent Systems) to successfully create applications that can work as rehearsal partners or teachers in the styles of jazz, *bossa nova* and other Brazilian genres. This line of research is strongly related to "sound and music analysis", since their results may feed accompaniment systems with patterns and rules that can be used to generate music.

2.4 Creativity Support Systems

Creativity as an asset is becoming critically important as enterprises, and individuals become more dependent on innovation to be competitive. Creativity Support Systems (CSSs) are processes or artifacts that improve productivity and empower users in creative processes. We are searching for new tools and systems that gives possibilities for the emergent roles in new system production and improves professional routine of creative tasks. Usually, CSSs are focused on the generation and orientation of new ideas (something like brainstorming techniques), but we take a broader view that incorporates the prevention of lock-in in the decision making and the design of user interfaces that do not constrain one's creative path. It includes, but is not limited to, development of prototyping and experimentation tools, evaluation of processes, development of new sensors and interfaces. We are also particularly interested in the use of artificial intelligence to generate creative systems, a field starting to emerge by the name of computational creativity.

2.5 Educational Systems

This point is a transversal axis of our research. Automatic accompaniment systems, rapid prototyping tools, new digital music instruments, they have all been used as learning tools. For some of them, however, education is the primary concern. For example, Daccord programs which teach how to play guitar, keyboard, and percussion, as well as robotics teaching experiments in partnership with RoboLivre. Our research in this field is, then, twofold: to create tools which enhance learning capabilities using innovative pedagogical approaches, and to develop metrics to evaluate them. One of the results of this research branch is the well-established company Daccord Music Software.

3 Illustrative Projects

In this section, we present illustrative projects developed by the members of MusTIC.

3.1 Previous Works

The same group, which became the MusTIC, performed some significant works. For example, the many systems headed by Giordano Cabral under the brand "Daccord." The first of these works, Daccord Guitar, was a full-featured accompaniment system, which led to a particular way of learning to play musical instruments, which brought lots of research problems: automatic synchronization, processing of lyrics and tablatures, intelligent calculation of chord positions and chord position paths, etc. The system was later supplemented by a Chord Dictionary, educational tools for guitar, keyboard, and drums. image: Daccord Violao

Figure 1: Daccord Guitar, um software educational de acompanhamento automática

Daccord Guitar started being developed in 1999 and became the initial product of the startup Daccord Music Software, still active today. The company continued to work in partnership with the scientific community and developed almost 70 music programs and music games, also generating four spin-offs, specializing on audio processing for security, educational technology, music games, and middleware for mobile devices.

This branch of works evolved into researches about automatic generation of rhythm, such as in the Cyber Joao (for guitar, by Marcio Dahia) and the CinBalada (for percussion, by Paulo Pereira and later Pablo Azevedo). On the other hand, Ernesto Trajano, Raphael Holanda, Didimo Junior, and Ricardo Scholz continued exploring the data structures of harmony and rhythm, performing analysis of microtiming, microdynamics, and functional harmony.

All these works were complemented by MSC researches on audio processing (Henrique Leão), harmony prediction (Sidney Cunha), collaborative music creation (Joao Paulo Rolim), and music visualization (Jarbas Jacome).

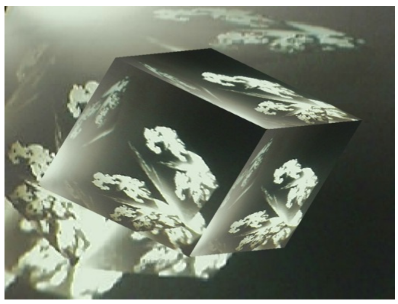


Figure 2: ViMus' Pseudo Fractal

The ViMus [1], for example, is an interactive system for embedded real-time visual processing. Interactive because both the artist and the audience can interact with the work. Real-time because the processing is done at the time that the result is being displayed. Moreover, integrated audiovisual and audio processing because that can be used to change the video, for example.

3.2 Audio Encoding and Sharing

Just before the boom of streaming services, Marcio Dahia performed extensive research about audio encoding, lead-

[1] https://jarbasjacome.wordpress.com/vimus/

ing to a new paradigm of music file-sharing based on layers. This way, services could download different resolutions according to their needs. This research was developed in the context of Canto Livre, an open-source music streaming platform, ordered by the Ministry of Culture, based on creative commons.

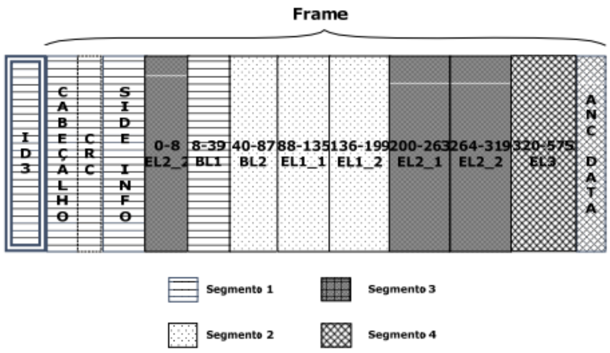


Figure 3: MP3 audio segmentation by perceptive significance

Members involved: Marcio Dahia, Geber Ramalho, Giordano Cabral, Silvio Meira

3.3 tAMARINO

tAMARINO [2] is a visual approach to rapid prototyping in physical computing. It proposes a unique and intuitive visual environment toolkit to accelerate physical computing prototypes both in the software and hardware fields.

Figure 4: Screenshot of tAMARINO's GUI

The evaluation reveals tAMARINO' success to straightforward, agile development – even on first-time prototyping – further lowering the time-to-market. This first version is designed for Arduino microcontrollers but is extendable to many other boards.

Members involved: Ricardo Brazileiro, Geber Ramalho, Abel Filho, João Paulo Cerquinho

3.4 Sketchument

DMI creation still requires a strong technical background. Based on the importance of prototyping in the process of designing things, Sketchument [3] is an environment devoted to helping non-technical users to quickly prototype DMIs, using multiple input modes and allowing the integration to other useful technologies [2].

[2] `https://vimeo.com/65594452`
[3] `https://vimeo.com/49199339`

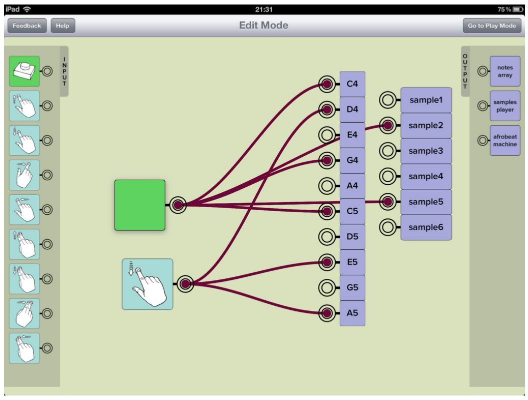


Figure 5: Sketchument on the iPad screen

From low-fi paper prototype to functional ones, passing through movies, questionnaires, interviews, Sketchument has been developed following the same prototyping philosophy we intend to propose to its users. The cyclic process of design-implementing-evaluating has produced valuable feedback from potential users, that have been very useful to back design choices and to push modifications.

Members involved: Filipe Calegario, Giordano Cabral, Geber Ramalho

3.5 LiVES

LiVES (LiVES Video Editing System) [4] development began in late 2002. The author was inspired to start creating a new video editing application after purchasing a new photo camera. In addition to taking photos, the camera was able to record small clips of video; however, for technological reasons, the video clips were limited to a duration of just ten seconds. As well as the limitation in duration, the clips were recorded without audio because of a lack of microphone on the camera.

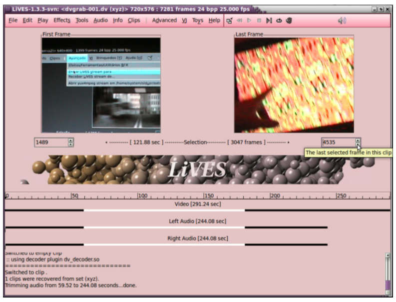


Figure 6: Screenshot of LiVES's GUI

The author decided that a means to increase the usefulness of the camera would be to use a program to join together several of these ten-second segments of video, and to add in some audio – perhaps music or commentary. Finally, he hoped to be able to encode the finished result. However, there was a problem – the author was committed to using only Linux, and at that time, none of the applications available for that operating system could import the

[4] `http://lives-video.com/`

camera's format. On the other hand, the author was able to use a different program to play the video clips on Linux, although not to edit them. Since the player program was able to output the frames as a sequence of images, the author had the idea of making a simple editor to play the images in sequence and to edit the images, then to add sound, and then to encode. Thus LiVES was born.

Members involved: Gabriel Finch (Salsaman), Giordano Cabral

3.6 Illusio

Illusio [5] is a new Digital Musical Instrument [3]. It is based on an augmented multi-touch interface that combines a traditional multi-touch surface and a device similar to a guitar pedal.

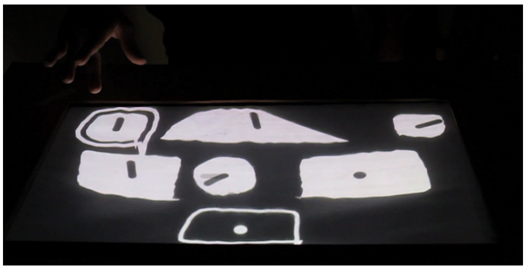


Figure 7: Illusio Multitouch Interface

Illusio allows users to perform by drawing and by associating the sketches with live loops. These loops are manipulated based on a concept called hierarchical live looping, which extends traditional live looping through the use of a musical tree, in which any music operation applied to a given node affects all its children nodes.

Members involved: Jerônimo Barbosa, Filipe Calegario, Geber Ramalho, Giordano Cabral, Veronica Teichrieb

3.7 Sensor Integration on Music

This project used a multisensor integration approach to evaluate gesture interface technologies, in particular, the Leap Motion, to capture gesture nuances in order to improve DMIs' expressivity and meet professional musicians demands obtained in previous research. We found that we were able to capture the nuances and meet latency guidelines from the literature, however instability with the Leap Motion "downward" setup caused system instability when both hands were inside the sensor's FOV.

Members involved: Eduardo Santos, Geber Ramalho

3.8 Meta Learning to Create Audio Classifiers

The group started working with audio processing and analysis in 1999, but it became significant research axes when Giordano Cabral came back from his Ph.D., in Music Information Retrieval for Interactive Systems. Some of the works deserve to be acknowledged, such as that of Dalton Francisco, dealing with search in a space of audio features and ML classifier parameters, that of Sidney Cunha, dealing with the enhancement of chord recognition (and transcription) methods, and that of José Menezes, about the use of multi-objective evolutionary computing methods to improve automatic feature extraction, such as in systems like the EDS.

This research led to innovations, such as those from a startup called Audio Alerta. Audio Alerta systems use audio analysis for public security. For example, one of the products attaches four microphones to monitoring cameras, and process the incoming audio to detect, classify and localize gunshot, explosions, and other alarming sounds.

Figure 8: One of Audio Alerta products

Members involved: Márcio Dahia, Dalton Francisco, Jose Menezes, Giordano Cabral

3.9 Audio API to Video Games

Another innovative use of MIR and audio analysis capabilities was the creation of the MusiGames Audio API. Created jointly with the startup MusiGames, it was a set of functions to automatically extract musical content from both audio files and the incoming audio to create new game experiences. For example, the API allowed the creation of games using the microphone of mobile phones as input, or the microphone of video game consoles. The API also permitted to convert a song into a game level automatically or to manipulate music the music content freely. An overall of 23 game titles was released using this API functionalities.

Figure 9: Some game titles which used the Audio API

[5]https://vimeo.com/25641586

Members involved: Giordano Cabral, Marcio Dahia, Roberto Cassio Jr

3.10 Brazyle and the Flow Machines Project

One of the significant scientific contributions of the group was its collaboration with SONY Computer Science Lab in Paris and the Flow Records in the context of the Flow Machines Project. Flow Machines are a state-of-the-art set of artificial intelligence tools to help compose and arrange music. The Brazilian counterpart inside the project was called Brazyle.

Members involved: Giordano Cabral.

Figure 10: the Brazyle project

3.11 Catalog to Makers

The CatalogToMakers [6] is a collaborative catalog of electronics components. Its main goal is to help enthusiasts of some areas of the physical computing to develop their prototypes that best way possible, centralizing the information about the components in a unique place and reducing the time of all creative process.

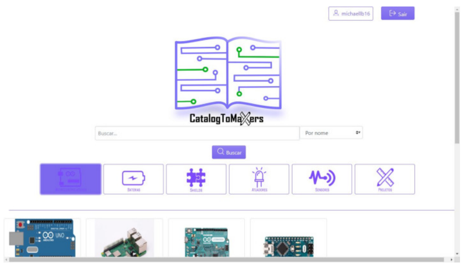


Figure 11: Screenshot of Catalog to Makers' GUI

Members involved: Michael Lopes, Filipe Calegario, Giordano Cabral

3.12 Vio.LED

Vio.LED [7] aims to identify usability guidelines on systems that benefit from IoT technology to improve efficiency in the musical instruments learning the process, mainly the acoustic guitar, and support novice musicians.

Within this project, the first step was the creation of a musical instrument containing LED light that can show music notes, chords, and scales. So, a user can

Figure 12: Vio.LED prototype

learn faster how to play particular songs. The second one was the automatic synchronization of music from streaming services to representations of this music (lyrics, chords, melody) from textual websites.

The research has, then, multiple facets — first, the development of music information retrieval techniques to integrate many music sources. Second, the development of hardware (augmented musical instruments). Third, the use of HCI and Design research tools to identify the usability guidelines to improve the software.

Members involved: Eduardo Santos, Giordano Cabral, Geber Ramalho

3.13 Cubmusic

CubeMusic [8] is an intelligent cube that works in conjunction with a smartphone/tablet app with Android operating system. Was developed with the Human-Computer Interaction discipline team offered at CIn-UFPE, through the initiative and idealization of the master's degree student in joining music to the challenge of building physical and digital toys.

Figure 13: Cubmusic Prototype

Members involved: Rute Maxsuelly, Giordano Cabral.

3.14 Technological resources and teaching strategies

The research is about strategies to bring the "female audience" closer to the technology area. Some approaches are maker culture, robotics, and game development.

Members involved: Mychelline Souto Cunha, Giordano Cabral.

[6] https://catalogtomakers.com.br/
[7] https://www.youtube.com/watch?v=qGpVHnzdWBg
[8] https://github.com/Rute123/CubeMusic

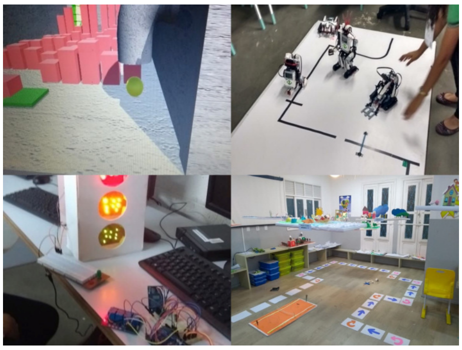


Figure 14: Game, Robotics, Maker Projects and Programming for Children

3.15 Bridge between Extended Reality and Music Interaction

It is a collaborative research between MusTIC and Voxar Lab to explore an Extended Reality interaction (Virtual and Augmented Reality). Principal working on the definition of a Natural Language to Human-Machine Interaction with a many of gesture contexts (like to body, hand, head, wand).

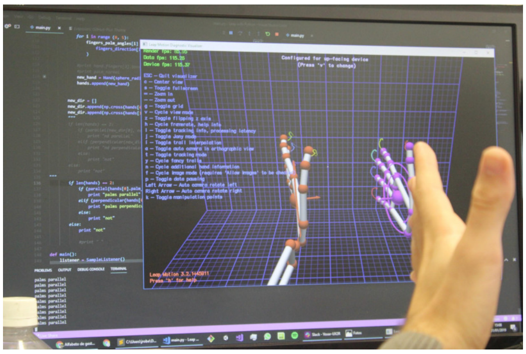


Figure 15: Prototype of Extended Reality and Music Interaction

Members involved: Jader Abreu, Giordano Cabral, Geber Ramalho

3.16 Personality Music

The project aims to understand the relationship of people's personality to their musical preferences using statistical analysis and machine learning techniques. As some of the results, we can mention Popularity x Extraversion, Valence x Openness, Danceability x Extraversion.

Members involved: Delando Júnior, Giordano Cabral

3.17 TumTá and Pisada

TumTá is a wearable Digital Dance and Music Instrument (DDMI) [4] in the form of a pair of insoles to be placed inside the shoes. It detects heel foot stomps and triggers samples from them. It was designed to give new sonic possibilities to the bold foot-stomping dance of Cavalo Marinho, a tradition from the Northeast of Brazil. It was developed as a demand from Helder Vasconcelos, a dancer, musician, and actor formed by this tradition, to explore new sound and dance possibilities in a performance he was devising [9].

Figure 16: Helder Vasconcelos playing/dancing with "TumTá" during his solo performance "FOCO"

It was designed with a handmade pressure sensor from conductive foam and thread that was rugged enough to receive bold stomps. These insoles were connected through wires to a wireless transmitter belt, that did not constrain the dancer's movement.

Pisada is another DDMI designed to fulfill some limitations of TumTá that did not afford the player to change the samples to be triggered during the performance. It consisted of a set of ten big and thin pads to be spread around the stage to change the sound banks. It represented the same functional qualities of a MIDI pedalboard being with a different physical structure that allowed broader foot-pressing gestures, so the performer could press it while dancing.

Figure 17: Some "Pisadas" spread on the floor

Members involved: João Tragtenberg, Helder Vasconcelos, Filipe Calegario, Giordano Cabral, Geber Ramalho

[9] `https://www.youtube.com/watch?v=m4q6iD513pY`

3.18 Giromin

Giromin is a wireless wearable free-gesture Digital Dance and Music Instrument [4]. It was made to be worn around the torso and on the upper arm not to impose any movement restrictions. It was designed to allow more expressive gestures in the control of continuous musical parameters on synthesizers (usually done with knobs and sliders). It afforded a precise instrumental control while allowing the performer to dance.

Figure 18: Giromin in action during the performance "Gira" in NIME 2019

It was motivated by research that suggested that the musical community in the Northeast of Brazil [5] saw that it was important for electronic musicians gestures to be perceivable by the audience and later used in the "Gira" performance. Each wearable module is composed of an accelerometer, gyroscope and magnetometer, which could extract movement information without imposing physical restrictions. It used a sensor fusion algorithm to extract orientation data of each limb together with rotation speed and acceleration data.

Members involved: João Tragtenberg, Filipe Calegario, Giordano Cabral, Geber Ramalho

3.19 Pandivá

Pandivá is an exploratory functional prototype which merged the guitar-inspired posture, the way of triggering sounds by tapping a tambourine skin and the way of altering the pitch using a trombone slide [5]. The instrument was called Pandivá (reduction of Portuguese words "*pandeiro de vara*", in English: slide tambourine).

Members involved: Filipe Calegario, Giordano Cabral, Geber Ramalho

3.20 Probatio

This project aims to address the following questions: how can we provide structured and exploratory paths to generate digital musical instruments (DMI) ideas? How can we decrease the time and effort involved in building functional DMI prototypes? To deal with these questions, we

Figure 19: Pandivá Prototype

developed a physical prototyping toolkit for building functional DMI prototypes: Probatio [10], a modular system of blocks and supports to prototype instruments based on specific ways of holding and gestural controls for musical interaction [6, 7]

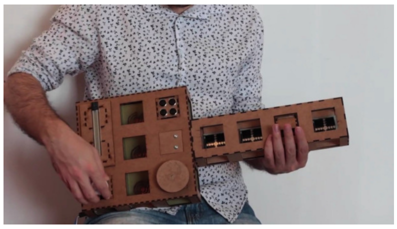


Figure 20: Probatio Prototype

This research was developed during Filipe Calegario's Ph.D. and since then has a continuous collaboration with the Input Devices and Musical Interaction Laboratory (IDMIL[11]), Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT[12]), McGill University, Canada, and Inria Lille, France.

Members involved: Filipe Calegario, Evandro Natividade, Giordano Cabral, Geber Ramalho

3.21 *Marine*

marine [13] is an open-source software focused in lowering the usability entry barrier for interactive poetics experimentation in performing arts, through the use of motion capture sensors to control lighting, image projections, sound, and connected objects [8].

The system is built over Processing 3+ and uses EyesWeb to compute movement features that, together with performers' position, can be used to control stage. Java developers may implement and publish new interactive behaviors as "*marine* elements," so that artists can import and configure them through a user interface. In short, *marine* proposes a UI paradigm shift for interactivity experimentation in performing arts, from the most used visual programming software - which requires programming

[10] `http://probat.io`
[11] `http://idmil.org`
[12] `http://cirmmt.org`
[13] `http://www.marineframework.org`

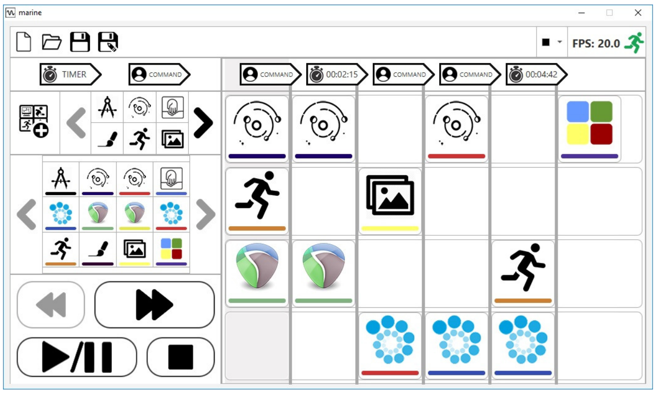


Figure 21: Screenshot of marine GUI

knowledge - to a higher level timeline oriented one - which does not.

Figure 22: *3 catástrofes sobre o prazer* using *marine* to control lighting, sound and projections

A formal evaluation of *marine* occurred through comparative tests and interviews with ten dancers. The results showed that *marine* improved effectiveness, efficiency, engagement and ease of learning, providing a more pleasant user experience when compared to visual programming software. So far, the system has been used in the performance *3 catástrofes sobre o prazer*[14] [15] [16] [17] and the interactive art installation *materia animata*[18].

Members involved: Ricardo Scholz, Geber Ramalho

4 A Cluster of Innovation in Music via Technology

The work of the MusTIC group led to many different initiatives (either startups or consolidated companies). Daccord Music Software, Batebit, LiVES, CESAR music-related projects, MusiGames Studio, Audio Alerta, ISI-TICs CIMUS, and a dozen of others were directly related to MusTIC.

Former and current researchers include Alexandre Braz, Dalton Araujo, Didimo Junior, Douglas Brito, Eduardo Santos, Erika Pessoa, Ernesto Trajano, Evandro Natividade, Filipe Calegario, Fulvio Figueroa, Gabriel Finch, Geber Ramalho, Giordano Cabral, Horhanna Oliveira, Hugo Santana, Izabel Zanforlin, Jader Abreu, Jarbas Jácome, Jaziel Vitalino, Jerônimo Barbosa, José Menezes, João Paulo Cerquinho, João Tragtenberg, Luca Dias, Luiz Delando Junior, Marcio Dahia, Mariane Marins, Michael Lopes, Mychelline Souto Cunha, Nélson Almeida, Paulo Sérgio Nunes, Pedro Aléssio, Raphael Holanda, Renato Celso, Ricardo Brazileiro, Ricardo Scholz, Rodrigo Medeiros, Rute Maxsuelly, Samuel Martins, Sidney Cunha, Sofia Galvão and Valter Jorge.

5 References

References

[1] Eduardo Reck Miranda and Marcelo M. Wanderley. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. A-R Editions, Middleton, 2006.

[2] Filipe Calegario, Jerônimo Barbosa, Geber Ramalho, Giordano Cabral, and Gabriel Finch. Sketchument: Empowering users to build DMIs through prototyping. *Organised Sound*, 18(03):314–327, 2013.

[3] Jerônimo Barbosa, Filipe Calegario, Veronica Teichrieb, Geber Ramalho, and Giordano Cabral. Illusio: A Drawing-Based Digital Music Instrument. *NIME '13 Proceedings of the 2013 Conference on New Interfaces for Musical Expression*, pages 499–502, 2013.

[4] João Tragtenberg, Filipe Calegario, Giordano Cabral, and Geber Ramalho. Towards the Concept of Digital Dance and Music Instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Porto Alegre, Brazil, 2019.

[5] Jerônimo Barbosa, Filipe Calegario, João Tragtenberg, Giordano Cabral, Geber Ramalho, and Marcelo M Wanderley. Designing DMIs for Popular Music in the Brazilian Northeast : Lessons Learned. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 277–280, 2015.

[6] Filipe Calegario. *Designing Digital Musical Instruments Using Probatio: A Physical Prototyping Toolkit*. Computational Synthesis and Creative Systems. Springer International Publishing, Cham, 2019.

[7] Filipe Calegario, Marcelo M. Wanderley, Stéphane Huot, Giordano Cabral, and Geber Ramalho. A Method and Digital Musical Instruments: Generating Ideas and Prototypes. *IEEE MultiMedia*, 24(1):63–71, 2017.

[8] Ricardo E P Scholz and Geber L Ramalho. Lowering the Usability Entry Barrier to Interactive Poetics Experimentation in Dance. *Interacting with Computers*, 31(1):59–82, jan 2019.

[14] solo 1: `http://youtu.be/El6dmWhvGZo`
[15] solo 2: `http://youtu.be/66NuQ9i2WbY`
[16] solo 3: `http://youtu.be/V9gY8ELjACg`
[17] solo 4: `http://youtu.be/owB2sz2b9XE`
[18] `https://youtu.be/-MHRwlLUobM`

LCM-UFRGS Research Group Report: What are we doing in Computer Music?

Marcelo S. Pimenta [1], Rodrigo Schramm [2], Marcelo Johann [1]

[1] Instituto de Informática, UFRGS
Caixa Postal 15064 – 90501-970 Porto Alegre, RS

[2] Departamento de Música, UFRGS

{mpimenta, johann}@inf.ufrgs.br ; rschramm@ufrgs.br

Abstract

In this paper, we present a historical overview and a brief report of the main recent activities at LCM (Laboratório de Computação Musical) of UFRGS (Universidade Federal do Rio Grande do Sul).

1. Introduction

LCM (Laboratorio de Computação Musical) is a multidisciplinary Computer Music research laboratory at UFRGS.

Permanent members of LCM are Marcelo Johann e Marcelo Pimenta (Informatics Institute, UFRGS) e Rodrigo Schramm (Music Department, UFRGS). We have also partners from others institutions from Brazil and around the world like Eduardo Miranda (University of Plymouth, UK) and Eloy Fritsch (Music Department, UFRGS), Evandro Miletto (Instituto Federal do Rio Grande do Sul - IFRS), Luciano V. Flores (QI Technology Faculty) e Leandro Costalonga (Universidade Federal do Espírito Santo – UFES), the last four partners having worked during MSc or-and PhD at LCM.

As with any multidisciplinary team, our work reflects the background and interests of group members, together or individually. In this article, we will first describe the history of the LCM and in the following section, we present a summary of recent works at LCM.

2. LCM: a little bit of history

Officially, the Computer Music research at LCM-UFRGS began in 1994 at the Institute of Informatics, mainly related to works on Artificial Intelligence and Music, with Eloi Fritsch and Fabio Beckenkamp. Soon, the team integrated IC students, masters and doctoral students, making the group growing in size and importance. The LCM group has been present since the first Brazilian Symposiums on Musical Computation - SBCM and is one pioneer of the area in Brazil.

After completing his PhD thesis in Computer Music at the UFRGS Computer Science graduate program in 2002, Eloi Fritsch became a lecturer of Electronic Music at the Department of Music of UFRGS and started to implement the Electronic Music Center (Centro de Música Eletrônica - CME) of the Arts Institute of UFRGS. The CME provides three electroacoustic music labs with the most advanced features for computer composition. Research and extension teams at CME prepared a series of acousmatic music concerts with the UFRGS speaker orchestra, presenting compositions in different theatres and auditoriums of the university.

In 1998, Marcelo Pimenta joined UFRGS and the LCM. Soon afterwards came Marcelo Johann and more recently Rodrigo Schramm. The profile of our permanent members is a sample of how much we value diversity. In recent years we have had excellent partners such as Marcelo Queiroz (University of São Paulo - USP), Flavio Schiavoni (Universidade Federal de São João del Rei – UFSJ) and the partners of the Ubiquitous Music group (see Section 3.2 below), Maria Helena de Lima, Victor Lazzarini (National University of Ireland, Maynooth) and Damián Keller (Universidade Federal do Acre – UFAC). The LCM has successfully achieved international credibility, being part of important collaborative projects and being honoured with the opportunity to host the International Conference on New Interfaces for Musical Expression – NIME 2019.

3. LCM recent works

The main motivation of the team is the creation of computer-based support for creativity. Creativity plays a crucial role in culture. Creative activities provide personal, social, and educational benefit, but creativity takes different forms at different times and in different places. For Plato, and later for the Romantics, for instance, creativity was an attribute (a gift) of certain mysteriously favored individuals. Today's consensus favors the view that creativity can be developed through education and opportunity that it can be an attribute of teams and groups as well as individuals, and that its social, cultural, and technological contexts matter [62].

Because of the breadth of use to which artists put different forms of digital technology, and because they typically are not steeped in conventional information technology approaches, artists' perspectives on tools and applications may provide valuable insights into the needs of other kinds of users of digital technology, that would not be obvious in a more narrowly focused technological context. Conversely, outcomes of computer science research may challenge artists to rethink their established assumptions and practices.

To a software developer, it might seem that the keys to technology-based creative practices are simply equipment and software— developing and providing access to standard, commercial IT tools for artists. This perspective is useful as far as it goes, and it can provide a good way to get started with IT, but in the long run, it is an insufficiently rich or flexible one. “We make our tools; then our tools make us”, we can claim, inspired by Marshall McLuhan [75].

Technology-based creative practices can constitute an important domain of research. It is inherently exploratory and inherently transdisciplinary. Concerned at its core with how people perceive, experience, and use information technology, it pushes on the boundaries of both digital technology and arts. In transdisciplinary research, the point is not just application of given methodologies but also a result of imagining entirely new possibilities for what disciplines can do.

3.1 Music creation by novices

Traditionally music composition process – for both composers and songwriters - assumes taking in account some musical knowledge (music theory, musical forms, techniques and skills in instrument playing and so on) and adopting some (either conventional or not) music notation to represent the music resulting of such process.

For novices, we prefer the generic term “music creation”, and the only thing we can assume for a music creation process by novices is that we can make no assumptions about a novice’s skills or knowledge. So, a simple replication of musician-oriented concepts, interfaces, symbols and features, without a careful analysis of their requirements and world views, could result in tools that would seem useless and unusable to these novice users. We intend to provide any user—either experienced musicians or not —access to meaningful and engaging musical experiences. Thus, we focus less on the musical quality of the finished work and more on the rich and flexible support for the music creation.

In previous work [79], we have proposed, discussed and illustrated some principles for music creation by novices in networked music environments: (1) music creation by novices should be prototypical; and (2) music creation by novices should be cooperative. These principles have emerged during CODES [76] design and development. CODES is a Web-based networked music environment designed to support cooperative ways of music creation by novices [77]. Some mechanisms and concepts related to the principle #2 - music creation by novices should be cooperative - can be found in [80].

Principle #1 is particularly interesting for guiding technology-based artistic creation by novices as a Design activity [73]. A design activity can be guided by cyclical prototypes construction where experimentation is not only allowed but also stimulated as well. Prototype creation allows designers to identify the error (or parts that are most likely to have problems), solve those errors/problems and then continue the creation, leading to unexpected discoveries and innovations that may or may not take the project beyond its initial scope. Prototypes are thus used to continuously revise and expand the design.

Design activity needs iteration (due its cyclic nature) and interaction design, taking ideally less time and effort to build prototypes. Indeed, the goal of interaction design is to create products that enable the user to achieve their objective(s) in the best way possible [82]. If this definition sounds broad, that’s because the field (Interaction Design and User Experience Design) is rather broad: the interaction between a user and a product often involves elements like aesthetics, motion, sound, space and much more. And the User Experience (UX) Design refers to shaping the experience of using a product, and most part of that experience involves some interaction between user and the product.

In our previous work CODES, creation of music prototypes was made possible by means a good interaction design, a visual notation with icons for representing sound and music, and direct manipulation metaphor for manipulating such icons [15,16,78]. Recently, this interface between Computer Music and Human-Computer Interaction (HCI) was called musical interaction design, having as goal “to bridge models and tools from both domains by combining recent developments, and providing a musical interaction design workbench to facilitate the exploration and definition of new interactive technologies for both musical creation and performance” [84].

From this perspective, the work of Evandro Miletto - first in the master's degree [8] and later in the doctorate [7] – investigated how networked music technology can provide adequate support for music creation and to discuss how it is possible to overcome a set of natural barriers and to define requirements specifically oriented to user novices in music. As the testbed of this research it was developed an environment for cooperative musical creation on the web, CODES [3,7,8,10,18,19, 47,48,49,50,51]. CODES (COoperative Musical Prototypes DESign) is a web-based environment designed and developed to allow cooperative musical experimentation by novices in order to turn them creators of musical content. Evandro is now a lecturer at IFRS and continues to do research in this environment. Many master's and undergraduate studies were carried out in the context of CODES - [14, 48, 54], such as the works of Aurélio Hoppe, Felipe Scheeren and Guilherme Testa.

3.2 Ubiquitous Music

We have proposed the adoption of the term ubiquitous music [12, 69] (or simply ubimus) to promote practices that empower participants of musical experiences through socially oriented, creativity-enhancing tools [28]. To achieve this goal, our group has been engaged in a multidisciplinary effort to investigate the creative potential of converging forms of social interaction, mobile and distributed technologies and innovative music-making practices.

Recently, the tendency for major increase in processing power, and convergence of technologies in everyday, ubiquitous consumer mobile devices has attracted the interest of several computer music researchers and artists. They saw the potential of combining music and mobile devices—something that is already being called mobile music [67]— in experiments that involve supporting mobility intrinsic to musicians, distributed and instant access to musical information and processing, design of new instruments (like [63]), new forms of audience participation and collaboration, localisation and context awareness (as in locative audio) and connectivity for musical activities [61, 68].

One of our goals is to develop tools which take advantage of these inclusive contexts, providing conditions to novices to participate in creative activities, ideally in any place and at any moment. Our strategy, for this, relies on repurposing everyday consumer mobile devices (devices they already own, and are familiar with) as ubiquitous music interfaces for use in musical activities, taking benefit from their distinctive capabilities of portability, mobility, and connectivity and above all from their availability to the average person (including novices in music) [35]. In fact, the smartphone has become the emblematic figure of ubiquitous computing and of Ubiquitous Music: it is the most popular and ubiquitous computing device that ever existed. Breaking such barrier that keep novices away from artistically expressing themselves, the next challenge is to investigate how to provide support for such artistic activities in a way they can be carried out even by people with little or no previous artistic knowledge.

This work started as a PhD thesis theme by Luciano Vargas Flores - An Infrastructure for the Design of Musical Interaction with Daily Mobile Devices [6]. Luciano's thesis was an exploratory research that investigated the possibilities and elements involved in the use of consumer mobile devices as tools for musical activities. Common mobile information and communication devices - mobile phones, smartphones and handheld computers - have increasingly incorporated functionality through the convergence phenomenon and tend to become generic mobile computing platforms. The related existing works presented specific solutions, requiring a proposal for the organization of general concepts and the basis for interaction design in this area. From the point of view of the Human-Computer Interaction area, a central problem in this context is how to perform the interaction design of a system for musical activities involving non-specific mobile devices, that is, devices were not originally designed for musical tasks. This is what we identify in this work as the problem of "device reuse" (device repurposing). The solution involves not modifying the devices, but rather finding their own and varied (alternative) ways of manipulating data and musical information using the features already on the devices. In addition, the design of musical interaction should not only consider the instrumental paradigm - based on the interaction with traditional musical instruments - but rather include in the design space other interaction paradigms, many of them originated and already adopted in the area of Computer Music. In his work, these various paradigms or forms of musical interaction were identified and formalized as proto-patterns of interaction. The collection and refinement of the patterns was possible through an exploratory methodology based on the experience of developing several mobile musical prototypes, which also included an extensive bibliographic review and the analysis of existing systems. The result of this process was organized in the form of an infrastructure composed of concepts and principles, interaction patterns and tools to support the design of interactive musical systems involving mobile consumer devices.

When we started working and exchanging ideas about this theme [35, 36, 30, 40, 29, 45], soon more partners emerged, and we formed a multidisciplinary and multi-institutional and international group (called g-ubimus). The meetings of this research group have been held since 2010 in the form of a workshop - Ubimus Workshop on Ubiquical Music, but which seems to have become effectively international: the VI Ubimus was held in Sweden (2015- Växjö) and the near Ubimus workshop will be a satellite event of the CMMR conference, to be held in France now in 2019. The repercussion of the group's work has been great and culminated with the publication of a book by Springer in 2014, called Ubiquitous Music [12], and of which Marcelo Pimenta is one of the editors.

More recently, in collaboration with Leandro Costalonga (UFES), we are investigating biological and cognitive principles for the adoption of musicality in the conception of Computer Music systems and tools and non-conventional instruments (DMIs, Digital Music Instruments).

3.3 Music Information Retrieval (MIR)

The LCM group have also integrated techniques based on Music Information Retrieval (MIR) into its research portfolio. Over the last decade, the growing number of machine learning algorithms have also motivated the development of new tools for automatic music understanding and music creation.

Multi-pitch detection is an essential task for the development of automatic music transcription systems. Algorithms based on spectrogram factorization have been investigated by the LCM team and applied to automatic music transcription of audio recordings of a cappella performances with multiple singers [58]. The research on this topic has been boosted with the implementation of a collaborative project started in 2016, connecting faculty members from Queen Mary University of London, University of Edinburgh and UFRGS. In the scope of this collaboration, we have proposed a system for multi-pitch detection and voice assignment that integrates an acoustic and a music language model. The acoustic model performs spectrogram decomposition, extending Probabilistic Latent Component Analysis (PLCA) using a 6-dimensional dictionary with pre-extracted log-spectral templates. The music language model performs voice separation and voice assignment using hidden Markov models that apply musicological assumptions. The system was evaluated with audio recordings of polyphonic vocal music, being able to detect multiple concurrent pitches and assign each detected pitch to a specific voice type such as soprano, alto, tenor or bass (SATB). A paper with results of this research received the Best Paper Award at the AES International Conference on Semantic Audio, Erlangen [59].

Variants of our PLCA based algorithm for multi-pitch detection were explored to build alternative tools for automatic transcription of musical instruments, as the diatonic harmonica and the electric guitar.

Since methods based on spectrogram factorization may suffer from local-optima issues in the presence of harmonic overlap or considerable timbre variability, we proposed [56] a set of harmonic constraints that are inherent to the Harmonica instrument note layout or are caused by specific diatonic Harmonica playing techniques. These constraints help to guide the factorization process of the PLCA based algorithm to meaningful convergence, resulting in better estimates of multi-pitch activations. This research is in progress, and it will make available a new audio dataset containing solo recordings of diatonic Harmonica excerpts and the respective multi-pitch annotations.

Aiming to support the creative musical process, the LCM team has integrated the MIR research with the development of new interfaces for musical expression (NIME). Such development has as premises the timbre replacement of acoustic instruments and also the possibility to use it as an embedded system. These two premises aim to address the rapid instrument augmentation, and the entire system was designed to run on a low-cost embedded computer, suitable for live performance and easy to customize for different use cases. A prototype is described in [57], where the core of the system implements real-time spectrum factorization, decomposing polyphonic audio input signals into music note activations. We have successfully implemented our system for the augmentation of electric guitars, where the timbre replacement is achieved by applying the extracted polyphonic pitch activations to mix a variety of synthetic or sampled sounds into the output audio signal.

3.3 Digital sound synthesis on low-cost platforms

Microcontrollers and other integrated processors have evolved significantly not only in terms of technology but also in their availability and easy of

use. One reference milestone, for instance, is the Arduino project, which helped to popularize simple standard microcontroller boards as well as a standard programming IDE.

At first such readily available low-cost boards were very easy to use for simple projects but lacked performance for more sophisticated DSP tasks such as audio synthesis and processing, because they employed 8-bit microprocessors without even hardware multiplications in 16 bits. But in the last years, several alternatives emerged using 32-bit cores with higher clock rates and integrated resources, with single cycle 32-bit hardware multiplication, such as the Arduino Due, Teensy 3.6, ESP32, for example.

In such processors, several interesting audio applications can be implemented, with standard high-quality sample rates and bit depths, external ADCs and DACs. It is important to note that this is still a much simpler and lower cost solution compared to the processing power available on integrated single board computers and mobile phones, which, by their turn, are still below a typical computational performance of a desktop or server processor, as show in table 1.

Platform / Processor	Speed
Intel Core i7	~3,5 GFLOPs/core
Qualcomm Snapdragon 810 (Sony Xperia Z5)	~1,45 GFLOPs/core
Raspberry Pi 3	~180 MFLOPs/core
Arduino Due	1,11 MFLOPs (float)
Arduino Uno	0,089 MFLOPs (float)

Table 1

Nevertheless, it might go unnoticed that the computing power used to implement several digital synthesizers and effect processors of the first generations in the past was also very modest. The flagship DX7 digital FM synthesizer from Yamaha was one of the highest selling synthesizers in history and together with other products they made the FM synthesis technology to be the second most profitable patent for the Stanford University before it expired. Its hardware sound generation engine used two custom VLSI circuits with fixed point arithmetic and a single hardware adder. Not even a multiplier was needed, as the developers wisely used exponent addition to represent multiplication and table-based conversions between linear values and their logarithms.

In other words, with current low-cost microcontroller boards, it is possible to implement several fully functional digital synthesizers and processors. We have been experimenting with a few basic instruments as a starting point. Firstly, an additive synthesizer was implemented [86], what showed that an organ with full 61-notes polyphony and 9 harmonics per key can be implemented on the 84 MHz Arduino Due, with external 16-bit DAC and basic amplitude and pitch modulations. After that, two other synthesizers were implemented. The first one is an FM synthesizer with the same resources as the original FM DX7, however restricted to only 6 notes of polyphony. The second one was a subtractive synthesizer composed of two oscillators using band limited audio generating algorithms, one digital resonant filter and two envelopes [85, 88]. There are other implementations in the same line, such as [87], and we believe that such works must be promoted.

Although it is common to implement standard well known processes such as additive, subtractive, FM synthesis, sample playback (or ROMplers), it is worthwhile to note that the true value of such readiness to implement audio processes lies one its potential to innovate, making custom instruments, new synthesis algorithms, to employ novel parameters and controllers, to integrate music intelligence with sound generation and so on.

The design of alternative digital instruments has been addressed by integrating the computational resources of smartphones. The SIBILIM project [60] is a low-cost musical interface built on a resonance box made of cardboard containing customized push buttons with no electronic circuit. Sound generation is provided by an external smartphone that interacts with the push buttons through its video camera. Each button is tracked in real-time by the mobile application, and the user controller is mapped to a set of MIDI notes or control parameters. The sound is generated through synthesis or sample playback and can be amplified with the help of a transducer, which excites the resonance box. The embedded computer vision algorithm allows the rapid reconfiguration of the buttons' layout with no need for hard rewiring to any electronic circuit. These features allow for quick instrument customization for different use cases, such as low-cost projects for schools or instrument building workshops. Our case study used the SIBILIM for music education, where it was designed to develop the consciousness of music perception and to stimulate creativity through exercises of short tonal musical compositions.

4. Final Considerations

One of the challenges of the LCM group is the exploratory investigation of how to integrate digital technology into research and artistic creativity (including multimodal sound, music, and image interactions) through theories, concepts, principles, technologies and tools with the goal of investigating the creative potential of converging forms of social interaction, mobile and distributed technologies, and materially grounded artistic practices.

In this paper, we presented a historical overview, the team and a brief report of the activities of the LCM (Laboratório de Computação Musical) of UFRGS. The lab started in pioneer times, and the team has today turned into a potential group of researchers from partner institutions in Brazil and abroad, carrying out studies connecting music and technology beyond the frontiers of the knowledge, and placing a high value on creativity.

However, some questions remain open, as for example: what can be done with technology - integrating research and art, and also adopting new paradigms of interaction - to provide support to the creativity of ordinary people? What kind of experience does this suggest? And what kinds of behavior of creation and listening? Are mobile devices good generic platforms for integrating digital technology with research and artistic creativity as they are good as interactive music performance devices? These questions and others, which arise every day, motivate the continuity of studies in the LCM.

References

[1] Honing, H.: The origins of musicality. The MIT Press, Cambridge, Massachusetts, USA (2018)

[2] Creativity-centred design for ubiquitous musical activities: Two case studies. Journal of Music, Technology and Education., v.5, p.195 - 222, 2012.

[3] Interfaces for Musical Activities and Interfaces for Musicians are not the Same: The Case for CODES, a Web-based Environment for Cooperative Music Prototyping. Intl Conf. on Multimodal Interfaces, 2007.

[4] Pimenta, M. et al.. Ubiquitous Music: A Manifesto, Prologue of Book Ubiquitous Music. Springer, ISSN 1868-0305, 2014.

[5] Pimenta, M. S.; Miletto, E.M.; Flores, L.V.; HOPPE, A. F. Cooperative mechanisms for networked music. Future Generation Computer Systems, v.27, p.100 - 108, 2011.

[6] Flores, L. V.. Uma infraestrutura para o design da interação musical com dispositivos móveis cotidianos. 2014. Tese (doutorado) – UFRGS/PPGC, Porto Alegre, 2014.

[7] Miletto, E. M.. CODES: an interactive novice-oriented web-based environment for cooperative musical prototyping. 2009. Tese (doutorado) – UFRGS/PPGC, Porto Alegre, 2009.

[8] Miletto, E. M.; Pimenta, M. S.; Vicari, R. M.; Flores, L. V.: CODES: a web-based environment for cooperative music prototyping. In: Organised Sound, Cambridge Vol.10, n. 3 (2005), p. 243-253, il.

[9] Pinheiro da Silva, F.; Keller, D.; Silva, E. F.; Pimenta, Marcelo S.; Lazzarini, V.. Criatividade Musical Cotidiana: Estudo Exploratório de Atividades Musicais Ubíquas. Música Hodie, v. 13, p. 64-79, 2013.

[10] Miletto, E. M.; Pimenta, M.; Bouchet, F.; Sansonnet, J-P.; Keller, D.. Principles for Music Creation by Novices in Networked Music Environments. Journal of New Music Research, v. 40:3, p. 205-216, 2011.

[11] Keller, D.; Flores, L. V.; Pimenta, M.; Capasso, A.; Tinajero, P.. Converging Trends Toward Ubiquitous Music. Journal of New Music Research, v. 40:3, p. 265-276, 2011.

[12] Keller, D.; Lazzarini, V.; Pimenta, M. S.: Ubiquitous music. Cham: Springer, c2014. xxviii, 153 p., il. (Computational music science).

[13] Fritsch, E. F.; Flores, L. V.; Miletto, E. M.; Vicari, R. M.; Pimenta, M. S.: Software musical e sugestões de aplicação em aulas de música. In: Ensino de música: propostas para pensar e agir em sala de aula. São Paulo: Moderna, 2003. p. 141-157.

[14] Pimenta, M. S.; Keller, D.; Miletto, E.; Flores, L. V.; Testa, G. G.. Technological Support for Online Communities Focusing on Music Creation: Adopting Collaboration, Flexibility and Multiculturality from Brazilian Creativity Styles. In: Azab, Nahed Amin. (Org.). Cases on Web 2.0 in Developing Countries. 1ed.Hershey, USA: IGI Global, 2013, v., p. 283-312.

[15] Miletto, E. M.; Pimenta, M. S.; Sansonnet, J.-P.; Bouchet, F.: Social music making on the web with CODES. In: Annual ACM Symposium on Applied Computing (25.: 2010 Mar.: Sierre, Switzerland). Proceedings of the 2010 Annual ACM Symposium on Applied Computing. Danvers: ACM, c2010. p. 876-877.

[16] Miletto, E. M.; Pimenta, M. S.; Hoppe, Aurelio F.; Flores, L. V.: Who are the web composers?. In: International Conference on Online Communities and Social Computing (3.: 2009 July: San Diego, CA). Proceedings, Berlin: Springer, c2009 p. 381-390.

[17] Hoppe, A. F.; Miletto, E. M.; Flores, L V.; Pimenta, M. S.: Cooperation in musical prototypes design. In:

International Conference on Computer Supported Cooperative Work in Design (13.: 2009 April: Santiago, Chile). Proceedings, New York: IEEE, c2009. p. 698-703.

[18] Miletto, E. M.; Pimenta, M. S.; Vicari, R. M.: Using Codes cooperative music prototyping and educational perspectives. In: International Computer Music Conference (2005 Sept.: Barcelona). Conference proceedings.: SuviSoft OY, 2005. p. 387-390, il.

[19] Miletto, E. M.; Pimenta, M. S.; Costalonga, L. L.; Vicari, R. M.: Using the web-based cooperative music prototyping environment CODES in learning situations. In: Intelligent tutoring systems. ITS proceedings of the 7. International Conference on Intelligent Tutoring Systems 2004, Maceió (Lecture notes in computer science, v. 3220), p. 833-835.

[20] Keller, D.; Silva, F. P. da; Giorni, B.; Pimenta, M. S.; Queiroz, M.: Marcação espacial: estudo exploratório. In: Simpósio Brasileiro de Computação Musical (13.: 2011 ago 31-03 set.: Vitória, ES). Anais do 13º Simpósio Brasileiro de Computação Musical [recurso eletrônico]. Porto Alegre: SBC, 2011. [12] f., il.

[21] Radanovitsck, E. A. A.; Keller, D.; Flores, L. V.; Pimenta, M. S.; Queiroz, M.: MixDroid: marcação temporal para atividades criativas. In: Simpósio Brasileiro de Computação Musical (13.: 2011 ago 31-03 set.: Vitória, ES). Anais do 13º Simpósio Brasileiro de Computação Musical. Porto Alegre: SBC, 2011.

[22] Flores, L. V.; Pimenta, M. S.; Miranda, E. R.; Radanovitsck, E. A. A.; Keller, D.: Patterns for the design of musical interaction with everyday mobile devices. In: Simpósio de Fatores Humanos em Sistemas Computacionais (9.: 2010 out: Belo Horizonte, BR-MG). Anais [recurso eletrônico]. [S.l.]: Sociedade Brasileira de Computação, [2010?]. p. 121-128.

[23] Flores, L. V.; Miletto, E. M.; Pimenta, M. S.; Miranda, E. R.; Keller, D.: Musical interaction patterns: communicating computer music knowledge in a multidisciplinary project. In: ACM International Conference on Design of Communication SIGDOC (28.: 2010 Sept. 27-29: São Carlos, BR-SP). Proceedings of the 28th ACM International Conference on Design of Communication. New York: ACM, c2010. p. 199-206, il.

[24] Miletto, E. M.; Flores, L. V.; Pimenta, M. S.; Santagada, L.; Jerome, R.: Interfaces for Musical Activities and Interfaces for Musicians are not the Same: The Case for CODES, a Web-based Environment for Cooperative Music Prototyping. In: The Ninth International ACM Conference on Multimodal Interfaces (ICMI 2007), 2007, Nagoya, Japan. Proceedings of Ninth International ACM Conference on Multimodal Interfaces (ICMI 2007). New York: ACM Press, 2007. v. 1. p. 201-207. (ICMC), 2010.

[25] Keller, D.; Pinheiro da Silva, F.; Giorni, B.; Pimenta, M. S.; Queiroz, M.: Marcação Espacial: Estudo Exploratório In: 13 Simpósio Brasileiro de Computação Musical (SBCM 2011), 2011, Vitória. Anais do Simpósio Brasileiro de Computação Musical (pen drive). Porto Alegre: SBC, 2011. v.1. p.1 - 30.

[26] Radanovitsck, E. A. A.; Keller, D.; Flores, L.; Pimenta, Marcelo S.; Queiroz M.: mixDroid: marcação temporal para atividades criativas. In: Simpósio Brasileiro de Computação Musical, SBCM 2011, 2011, Vitória. Anais do Simpósio Brasileiro de Computação Musical (pen-drive). Porto Alegre: SBC, 2011. v.1.

[27] Yi, S.; Lazzarini, V.; Timoney, J.; Keller, D.; Pimenta, M. S. The mobile Csound platform. In: International Computer Music Conference ICMC 2012, 2012, Ljubljana. Proceedings of International Computer Music Conference ICMC 2012. Ann Arbor: Ann Arbor, MI: MPublishing, University of Michigan Library 2012, 2012. v.2012.

[28] Keller, D.; Pinheiro da Silva, F.; Silva, E. F.; Lazzarini, V.; Pimenta, M. S.: Opportunistic Design of Ubiquitous Music Systems: the impact of anchoring as creativity support) (In Portuguese, Design oportunista de sistemas musicais ubíquos: O impacto do fator de ancoragem no suporte à criatividade. In: Proceedings of 14th Brazilian Symposium on Computer Music (SBCM 2013), 2013, Porto Alegre: SBC, 2013. v.1. p.1

[29] Keller, D.; Timoney, J.; Costalonga, L.; Capasso A.; Tinajero, P; Lazzarini, V.; Pimenta, M. S.; De Lima, M. H.; Johann, M.: Ecologically grounded multimodal design: the Palafito 1.0 study. In: Proc. of 40th International Computer Music Conference (ICMC 2014) joint with the 11th Sound & Music Computing conference (SMC 2014)., 2014, Athens, Greece.

[30] Keller, D.; Otero, N.; Lazzini, V.; Pimenta, M. S.; Lima, M. H. de; Johann, M.; Costalonga, L.: Relational Properties in Interaction Aesthetics: The Ubiquitous Music Turn In: Electronic Visualisation and the Arts 2014, 2014.

[31] Keller, D.; Timoney, J.; Costalonga, L.; Capasso. A.; Tinajero, P.; Lazzarini, V.; Pimenta, M. S.; Lima, M. H. de; Johann, M.: Ecologically grounded multimodal design: the Palafito 1.0 study. In: Proceedings of 40th International Computer Music Conference (ICMC 2014) joint with the 11th Sound & Music Computing Conference (SMC 2014), 2014, Athens, Greece.

[32] Behrendt, F.: From calling a cloud to finding the missing track: Artistic approaches to mobile music. In: 2nd International Workshop on Mobile Music Technology. Vancouver, Canada (2005)

[33] Carroll, J. (ed.) Creativity and Rationale - Enhancing Human Experience by Design Human–Computer Interaction Series, 2013, VIII, 456p. ISBN 978-1-4471-4110-5.

[34] Essl, G.; Rohs, M.: Interactivity for Mobile Music-Making, Organised Sound 14(2): 197–207, 2009

[35] Flores, L.V., Pimenta, M.S., Keller, D.: Patterns for the design of musical interaction with everyday mobile devices. In: Proc. of the 9th Brazilian Symposium on Human Factors in Computing Systems. Belo Horizonte, Brazil (2010), 2010. v.1. p.41 – 50

[36] Flores, L.V.; Miletto, E. M.; Pimenta, M. S.; Miranda, E. R.; Keller, D.: Musical interaction patterns: communicating computer music knowledge in a

multidisciplinary project. In: Proceedings of the 28th Annual International Conference on Design of Communication, SIGDOC 2010, Sao Carlos, ACM, 2010. p.199 – 206

[37] Flores, L.V.; Pimenta, M. S.; Miranda, E. R.; Radanovitsck, E. A.; Keller, D.: Patterns for the Design of Musical Interaction with Everyday Mobile Devices In: (IHC 2010) IX Brazilian Symposium on Human Factors in Computing Systems, 2010, Belo Horizonte, v.1. p.41 – 50

[38] Gaye, L.; Holmquist, L.E.: Mobile music technology: Report on an emerging community. In: NIME 06: Proceedings of the 2006 Conference on New Interfaces for Musical Expression, pp. 22–25 (2006)

[39] Holmquist, L.E.: Ubiquitous music. Interactions 12(4), 71 ff. (2005). http://doi.acm.org/10.1145/1070960.1071002

[40] Keller, D.; Flores, L.V.; Pimenta, M.; Capasso, A.; Tinajero, P.: Converging Trends Toward Ubiquitous Music. Journal of New Music Research, v. 40:3, p.265 - 276, 2011. [http://www.tandfonline.com/doi/pdf/10.1080/09298215.2011.594514

[41] Keller, D.; Otero, N.; Lazzarini, V.; Pimenta, M. S.; Lima, M.H. de; Johann, M.; Costalonga, L.: Interaction Aesthetics and Ubiquitous Music In: Creativity in the Digital Age.1 ed. London: Springer London, 2015, v.1, p. 91-105. ISBN: 9781447166801,

[42] Keller, D.; Lazzarini, V.; Pimenta, M. S.: Ubiquitous Music. Heidelberg: Springer Intl. Publishing Switzerland, 2014, v.1. p.153.

[43] Keller, D.; Queiroz, M.; Barreiro, D.L.; Pimenta, M. S.: Anchoring in Ubiquitous Musical Activities In: International Computer Music Conference (ICMC), 2010, New York.

[44] Lazzarini,, V.; Keller, D.; Pimenta, M. S.: Prototyping of Ubiquitous Music Ecosystems. Journal of Cases on Information Technology., v.17, p.73 - 85, 2015.

[45] Lima, M.H.; Keller, D.; Pimenta, M.S.; Lazzarini, V.; Miletto, E.M.: Creativity-centred design for ubiquitous musical activities: Two case studies. J. Music Technol. Educ. 5(2), 195–222 (2012). doi:10.1386/jmte.5.2.195_1.

[46] McLuhan, M.; 1954, Notes on the Media as Art Forms. Explorations 2 (April): 6-13.

[47] Miletto, E. M.; Pimenta, M. S.; Vicari, R. M.; Flores, Codes, L.V.: a Web-based environment for cooperative music prototyping. Organised Sound (Print)., v.10, p.243 - 253, 2005.

[48] Miletto, E. M.; Pimenta, M.; Hoppe, A. F.; Flores, L.V.: Who Are the Web Composers? In: Proc. of 13th International Conference on Human-Computer Interaction (HCII 2009), 2009, San Diego, Berlin / Heidelberg: Springer, 2009. v.12. p.381 – 390

[49] Miletto, E.M.; Flores, L.V.; Pimenta, M.S.; Rutily, J.; Santagada, L.: Interfaces for musical activities and interfaces for musicians are not the same: The case for codes, a web-based environment for cooperative music prototyping. In: 9th International Conference on Multimodal Interfaces (ICMI), pp. 201–207. ACM, Nagoya (2007)

[50] Miletto, E.M.; Pimenta, M.S.; Bouchet, F.; Sansonnet, J.P.; Keller, D.: Principles for music creation by novices in networked music environments. J. New Music Res. 40(3), 205–216.

[51] Pimenta, M. S.; Miletto, E.M.; Flores, L.V.; Hoppe, A. F.: Cooperative mechanisms for networked music. Future Generation Computer Systems., v. 27, p.100 - 108, 2011.

[52] Pimenta, M.; Keller, D.; Lazzarini, V.: Prologue - Ubiquitous Music: A Manifesto, In: Keller, D.; Lazzarini, V.; Pimenta, M. S.: Ubiquitous Music. Heidelberg: Springer Intl.Publishing Switzerland, 2014, v.1, p.153

[53] Rogers, Y.; Sharp, H.; Preece, J.: Interaction Design – Beyond Human-Computer Interaction, Wiley, 3rd ed, 2011. ISBN 978-0-470-66576-3.

[54] Scheeren, F.; Pimenta, M. S.; Keller, D.; Lazzarini, V.: Coupling Social Network Services and Support for Online Communities in CODES Environment. In: 14th International Society for Music Information Retrieval Conference (ISMIR 2013), 2013, Curitiba.

[55] Wanderley, M. et al.: Human Computer Interaction meets Computer Music: The MIDWAY Project. Workshop Music and HCI, in conjunction with ACM CHI 2016, pp.4.

[56] Lins, F.; Johann, M.; E. Benetos; Schramm, R.: Automatic Transcription of Diatonic Harmonica Recordings, ICASSP, Brighton, UK, 12 May, 2019.

[57] Schramm, R.; Visi, F.; Brasil, A.; Johann, M.: A polyphonic pitch tracking embedded system for rapid instrument augmentation. NIME, Blacksburg, Virginia, USA, 4 Jun, 2018.

[58] Schramm, R.; McLeod, A.; Steedman, M; Benetos, E.: Multi-pitch detection and voice assignment for a capella recording of multiple singers. 18th International Society for Music Information Retrieval Conference, Suzhou, p552-559, 24 Oct, 2017.

[59] Schramm, R.; Benetos, E.: Automatic transcription of a cappella recordings from multiple singers. AES International Conference on Semantic Audio (BEST PAPER AWARD), Erlangen, 23 Jun, 2017.

[60] Nunes, H. de S.; Visi, F.; Coelho, L. H. W.; Schramm, R.: SIBILIM: a low-cost customizable wireless musical interface. NIME, Porto Alegre, 3 Jun, 2019.

[61] Behrendt, F.: From calling a cloud to finding the missing track: Artistic approaches to mobile music. In: 2nd International Workshop on Mobile Music Technology. Vancouver, Canada (2005)

[62] Carroll, J. (ed.): Creativity and Rationale - Enhancing Human Experience by Design Human–

Computer Interaction Series, 2013, VIII, 456p. ISBN 978-1-4471-4110-5.

[63] Essl, G.; Rohms, M.: Interactivity for Mobile Music-Making, Organised Sound 14(2): 197–207, 2009

[64] Flores, L.V.; Pimenta, M.S.; Keller, D.: Patterns for the design of musical interaction with everyday mobile devices. In: Proceedings of the 9th Brazilian Symposium on Human Factors in Computing Systems. Belo Horizonte, Brazil (2010), 2010. v.1. p.41 - 50

[65] Flores, L.V.; Miletto, E. M.; Pimenta, M. S.; Miranda, E. R.; Keller, D.: Musical interaction patterns: communicating computer music knowledge in a multidisciplinary project In: Proceedings of the 28th Annual International Conference on Design of Communication, SIGDOC 2010, Sao Carlos, ACM, 2010. p.199 - 206

[66] Flores, L.V.; Pimenta, M. S.; Miranda, E. R.; Radanovitsck, E. A.; Keller, D.: Patterns for the Design of Musical Interaction with Everyday Mobile Devices In: (IHC 2010) IX Brazilian Symposium on Human Factors in Computing Systems, 2010, Belo Horizonte, v.1. p.41 – 50

[67] Gaye, L., Holmquist, L.E.: Mobile music technology: Report on an emerging community. In: NIME 06: Proceedings of the 2006 Conference on New Interfaces for Musical Expression, pp. 22–25 (2006)

[68] Holmquist, L.E.: Ubiquitous music. Interactions 12(4), 71 ff. (2005). http://doi.acm.org/10.1145/1070960.1071002

[69] Keller, D.; Flores, L.V.; Pimenta, M.; Capasso, A.; Tinajero, P.: Converging Trends Toward Ubiquitous Music. Journal of New Music Research., v.40:3, p.265 - 276, 2011. [http://www.tandfonline.com/doi/pdf/10.1080/09298215.2011.594514

[70] Keller, D.; Otero, N.; Lazzarini, V.; Pimenta, M. S.; Lima, M.H.; Johann, M.; Costalonga, L.: Interaction Aesthetics and Ubiquitous Music In: Creativity in the Digital Age.1 ed.London : Springer London, 2015, v.1, p. 91-105. ISBN: 9781447166801,

[71] Keller, D.; Lazzarini, V.; Pimenta, M. S.: Ubiquitous Music. Heidelberg: Springer Intl. Publishing Switzerland, 2014, v.1. p.153.

[72] Keller, D.; Queiroz, M.; Barreiro, D.L.; Pimenta, M. S.: Anchoring in Ubiquitous Musical Activities In: International Computer Music Conference (ICMC), 2010, New York.

[73] Lazzarini,, V.; Keller, D.; Pimenta, M. S.: Prototyping of Ubiquitous Music Ecosystems. Journal of Cases on Information Technology., v.17, p.73 - 85, 2015.

[74] Lima, M.H.; Keller, D.; Pimenta, M.S.; Lazzarini, V.; Miletto, E. M.: Creativity-centred design for ubiquitous musical activities: Two case studies. J. Music Technol. Educ. 5(2), 195–222 (2012). doi:10.1386/jmte.5.2.195_1.

[75] McLuhan, M.: 1954, Notes on the Media as Art Forms, Explorations 2 (April): 6-13.

[76] Miletto, E. M.; Pimenta, M. S.; Viccari, R. M.; Flores, L.V.: CODES: a Web-based environment for cooperative music prototyping. Organised Sound (Print)., v.10, p.243 - 253, 2005.

[77] Miletto, E. M.; Pimenta, M.; Hoppe, A. F.; Flores, L.V.: Who Are the Web Composers? In: Proc. of 13th International Conference on Human-Computer Interaction (HCII 2009), 2009, San Diego, Berlin / Heidelberg: Springer, 2009. v.12. p.381 – 390

[78] Miletto, E.M.; Flores, L.V.; Pimenta, M.S.; Rutily, J.; Santagada, L.: Interfaces for musical activities and interfaces for musicians are not the same: The case for codes, a web-based environment for cooperative music prototyping. In: 9th International Conference on Multimodal Interfaces (ICMI), pp. 201–207. ACM, Nagoya (2007)

[79] Miletto, E.M.; Pimenta, M.S.; Bouchet, F.; Sansonnet, J.P.; Keller, D.: Principles for music creation by novices in networked music environments. J. New Music Res. 40(3), 205–216. (2011). doi:10.1080/09298215.2011.603832

[80] Pimenta, M. S.; Miletto, E.M.; Flores, L.V.; Hoppe, A. F.: Cooperative mechanisms for networked music. Future Generation Computer Systems., v.27, p.100 - 108, 2011.

[81] Pimenta, M.; Keller, D.; Lazzarini, V.: Prologue - Ubiquitous Music: A Manifesto, In: [11]

[82] Rogers, Y.; Sharp, H.; Preece, J.: Interaction Design – Beyond Human-Computer Interaction, Wiley, 3rd ed, 2011. ISBN 978-0-470-66576-3.

[83] Scheeren, F.; Pimenta, M. S.; Keller, D.; Lazzarini, V.: Coupling Social Network Services and Support for Online Communities in CODES Environment. In: 14th International Society for Music Information Retrieval Conference (ISMIR 2013), 2013, Curitiba.

[84] Wanderley, M. et al.: Human Computer Interaction meets Computer Music: The MIDWAY Project. Workshop Music and HCI, in conjunction with ACM CHI 2016, pp.4.

[85] Rodolfo P. P.; Johann, M.; Pimenta, M.: A modular platform for a subtractive synthesizer on Arduino Due. Proceedings of 15th Braziliam Symposium on Computer Music, 2015.

[86] Johann, M.: An additive synthesis organ with full polyphony on Arduino Due. VI Ubimus. Växjö, Linnaeus University, Sweden, 2015.

[87] Cyrille H.: OCS-2: A Digital, Semi Modular Synthesizer. 2017. Available at: <http://nozoid.com/ocs-2/>

[88] Rodolfo P. P.; Johann, M.; Pimenta. M.: Design and implementation of an open-source subtractive synthesizer on the Arduino Due platform. In: SBCM 2017.

Computer Music Research Group - IME/USP Report for SBCM 2019

Fábio Gorodscy [1] , Guilherme Feulo [1] , Nicolas Figueiredo[1] ,
Paulo Vitor Itaboraí[1] , Roberto Bodo[1] , Rodrigo Borges[1] , Shayenne Moura[1]

[1]FLOSS Competence Center (CCSL), Room 118 – Department of Computer Science /
Institute of Mathematics and Statistics (IME) / University of São Paulo
Rua do Matão, 1010, CCSL – 05508-090 Sao Paulo, SP

compmus.ime@gmail.com

Abstract. *The following report presents some of the ongoing projects that are taking place in the group's laboratory. One of the noteable characteristics of this group is the extensive research spectrum, the plurality of research areas that are being studied by it's members, such as Music Information Retrieval, Signal Processing and New Interfaces for Musical Expression.*

1 Introduction

This report presents the Computer Music Research Group, part of the Department of Computer Science at the Institute of Mathematics and Statistics at the University of Sao Paulo (IME-USP). The group is coordinated by Prof. Dr. Marcelo Queiroz, and is composed of undergraduate, masters and PhD candidates. Its research covers many diverse topics on MIR (singing voice detection, query-by-humming, audio fingerprinting), signal processing (adaptive multi-resolution analysis), physical modeling and augmented instruments, amongst other topics under the computer music area. The group organizes seminars about it's members ongoing research or invited speakers (all recorded and available at `http://compmus.ime.usp.br/en/seminars`) and weekly open meetings in order to update the members about ongoing research and discuss articles and collaborations.

2 Ongoing Projects

2.1 An exploratory work in query-by-humming

Query-by-humming is a common topic in music information retrieval. In the query-by-humming task a hummed record representing imprecisely a target melody, is given to an application which is supposed to retrieve information about the target melody from a dataset. One algorithm addressing the task has to handle deviations in both time and frequency domains.

Fábio Goródscy reviews standard techniques presented in the academic literature and in commercial applications. Algorithms presented in the international conference of music information retrieval are reviewed, as well as the commercial application Soundhound, which are explored and tested. This work compares the performance of several strategies for query-byhumming within a unified query dataset.

Most of the concepts used througout the work can be found in [1]. Related work are [2][3][4][5].

The main goal of this work is to undercover the difficulties in mesuring similarity in this context, by comparing the performance of a commercial tool and also several alternative strategies.

2.2 Development of an efficient adaptive transform for music signals

Music signals can present very heterogeneous spectral characteristics, such as sharp attacks, long stationary tones, vibrato and tremolo all spread across the hearing frequency range. In such cases, fixed resolution spectrograms or even frequency-dependent resolution spectrograms (such as CQT [6]) may not result in a satisfactory representation of the signal. This is the motivation behind adaptive transforms, operators that utilize information about the signal being analyzed to compose a representation that prioritizes frequency or time resolution depending on the signal's characteristics at a given time and frequency band.

These types of algorithms are usually cost-intensive. The project currently in development as Nicolas Figueiredo's Masters thesis is the development of a low-cost adaptive transform algorithm that does not follow the traditional framework of most adaptive transforms [7, 8]. Instead of comparing between different representations (for example, STFTs calculated using 1024, 2048 and 4096-sample analysis windows) and choosing the best one for each time-frequency split, this algorithm uses bandpass filtering and undersampling [9] to isolate "interesting regions" of a given spectrogram and analyze them cheaply in greater detail. The main objectives of this project are to develop an adaptive transform whose computing cost is similar to other representations usually used in MIR tasks, and evaluate it against other multi-resolution and adaptive transforms according to their computing costs and sparsity of the resulting representation.

2.3 Using Active Acoustics techniques in musical instruments and art installations

Active Acoustics is a term used in the New Interfaces for Musical Expression (NIME) research field to describe the usage of sound and vibration inducing devices to drive electronic sounds into physical surfaces [10]. The result of inducing synthesized sounds into a complex sound radiating source can be vastly explored by artists and music technologists.

Using Active Acoustics in order to augment traditional musical instruments is an active topic in the NIME field. Nicolas Figueiredo and Paulo Itaboraí are currently reviewing the augmented active acoustic instruments's literature in order to develop an augmented banjo. They are currently testing E.Berdahl's results of PID control on an string [11] and trying to empirically expand these results to a banjo's membrane using a BELA Board (originally called BeagleRT [12]), a piezoelectric sensor and a sound transducer.

Another possible application is to explore the acoustical properties (resonances, formants, non-linearities) of physical plates made of different materials to naturally distort and filter the electronic sounds. Paulo Itaboraí is using actuated ceramic plates to expand electroacoustic diffusion systems for acousmatic music performance. Each ceramic piece has an independent audio channel and provides an unique distortion to the sound. This enables electroacoustic composers to embed part of the sound object elaboration and distortion into spatialization gestures. Paulo, in collaboration with the composer Alex Buck, proposed and presented a sound installation called "Acting Voices - Madrigale a Sei Vasi' in the 19th International Conference on New Interfaces for Musical Expression

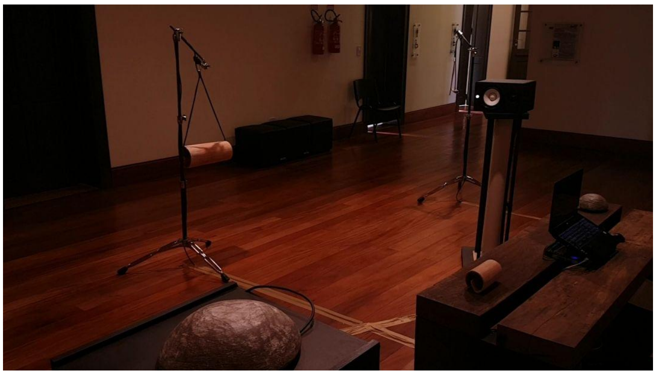


Figure 1: Installation "Active Voices - Madrigale a Sei Vasi" at NIME 2019 conference

2.4 Content Base Music Recommendation Systems

Music recommender systems (MRS) help users interacting with big digital song collections. They operate analyzing information about user's past behavior when listening to music, and suggest to each one of them the next song, album or artist to be heard. The most popular approach for implementing MRS is Collaborative Filtering (CF) [13], which associates each user to a listening profile, and assumes that similar profiles share musical preferences. According to the songs similar users interacted with, it estimates the probability of each unheard song being heard in order to make individual suggestions.

Rodrigo ir working towards one of main weakness in CF based solutions, usually described as the "cold start" problem: when new songs are included in the platform and need to be incorporate in the algorithm even without having any historical data. We propose a solution that associates the acoustic information extracted directly from the songs with user's preference [14]. In the case when there is a strong pattern in the content of the songs an user have heard so far, then it should be possible to recommend to him/her a new song that matches to his preference.

2.5 Experiments on Singing Voice Detection

Singing voice Detection in polyphonic audio signals is the problem that deals with determining which segments of a musical signal (with several sound sources) contain singing voice. This is an active topic in the Music Information Retrieval (MIR) field and has various applications, including automatic singer recognition [15], singing voice separation [16] and melody extraction [17].

Shayenne Moura started working with melody and accompaniment separation [18] and then focused her work on Singing Voice Detection (SVD), also referred as Vocal Detection. Her research is about how the SVD systems were developed in the past and what are the challenges remaining for this task [19]. She is evaluating the impact of using engineered descriptors in comparison with deep embeddings as features on the classification accuracy [20]; also, doing experiments with different mixes from the same pieces to evaluate the vocal detector under these constraints.

2.6 A framework for obtaining Musical Similarity measures

The spread of digital music allowed the appearance of datasets with millions of files. The processing of this huge number of audio files is carried out with techniques of Music Information Retrieval (MIR).

The main goal of this work is to study Musical Similarity which is to determine quantitatively how similar any two given songs are. The concept of musical similarity is subjective and there is no definition of general musical similarity. Therefore, the problem is addressed from similarities of individual musical elements, for instance, melody, harmony, tempo, metric, timbre, etc.

Roberto Bodo is working to reach this goal. He will review several measures of similarity computed between alternative representations of audio files (called audio features). With this, we can determine which songs are closest to each other, within a dataset. From the selected literature, several combinations were identified of audio features and similarity measures. At the current stage of the project, we have implemented a modular framework for obtaining similarity measures based on several features, aggregation strategies and distance models: we handled three types of similarity (timbristic, melodic, and rhythmic), and we calculated similarity matrices for a considerable number of datasets openly available.

In the future, we will explore the replacement of full songs by segments of them, analyze the obtained results and check if they extrapolate to datasets of world music. In addition, we will use deep learning techniques to learn which parts of the songs optimize the quality of the results, and create thumbnails extractors from the trained neural networks.

References

[1] meinard müller. *fundamentals of music processing: audio, analysis, algorithms, applications*. 2015.

[2] Justin Salamon, Joan Serrà, and Emilia Gómez. Tonal representations for music retrieval: from version identification to query-by-humming. *Int. J. of Multimedia Info. Retrieval, special issue on Hybrid Music Info. Retrieval*, 2(1):45–58, Mar. 2013.

[3] Ernesto López, Martín Rocamora, and Gonzalo Sosa. Búsqueda de música por tarareo, 2004.

[4] Bartłomiej Stasiak. Follow that tune – adaptive approach to dtw-based query-by-humming system. *ARCHIVES OF ACOUSTICS*, 39(4):467–476, 2014.

[5] Alexios Kotsifakos, Panagiotis Papapetrou, Jaakko Hollmen, Dimitrios Gunopulos, Vassilis Athitsos, and George Kollios. Hum-a-song: A subsequence matching with gaps-range-tolerances query-by-humming system. 5(10), 2012.

[6] Judith C Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[7] Alexey Lukin and Jeremy Todd. Adaptive time-frequency resolution for analysis and processing of audio. In *Audio Engineering Society Convention 120*. Audio Engineering Society, 2006.

[8] Florent Jaillet and Bruno Torrésani. Time-frequency jigsaw puzzle: Adaptive multiwindow and multilayered gabor expansions. *International Journal of Wavelets, Multiresolution and Information Processing*, 5(02):293–315, 2007.

[9] Rodney G Vaughan, Neil L Scott, and D Rod White. The theory of bandpass sampling. *IEEE Transactions on signal processing*, 39(9):1973–1984, 1991.

[10] Otso Lähdeoja. Active acoustic instruments for electronic chamber music. 2016.

[11] Edgar Joseph Berdahl. *Applications of feedback control to musical instrument design*. Stanford University, 2010.

[12] Andrew McPherson and Victor Zappi. An environment for submillisecond-latency audio and sensor processing on beaglebone black. In *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[13] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.

[14] Rodrigo Borges and Marcelo Queiroz. Automatic music recommendation based on acoustic content and implicit listening feedback. *Revista Música Hodie*, 18(1):31 – 43, jun. 2018.

[15] A. L. Berenzweig, D. P. W. Ellis, and S. Lawrence. Using voice segments to improve artist classification of music. In *22nd Int. Conf.: Virtual, Synthetic, and Entertainment Audio*. Audio Engineering Society, 2002.

[16] Yipeng Li and DeLiang Wang. Separation of singing voice from music accompaniment for monaural recordings. Technical report, Ohio State University Columbus United States, 2005.

[17] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, Aug. 2012.

[18] Shayenne Moura and Marcelo Queiroz. Melody and accompaniment separation using enhanced binary masks. In *Proceedings of the 16th Brazilian Symposium on Computer Music*, pages 164 – 165, São Paulo, 2017.

[19] Kyungyun Lee, Keunwoo Choi, and Juhan Nam. Revisiting singing voice detection: a quantitative review and the future outlook. In *19th Int. Soc. for Music Info. Retrieval Conf.*, Paris, France, 2018.

[20] Shayenne Moura. Singing voice detection using vggish embeddings. *19th International Society for Music Information Retrieval Conference, Paris, France*, 2018.

Posters

A cluster analysis of benchmark acoustic features on Brazilian music

Leonardo Antunes Ferreira , Estela Ribeiro , Carlos Eduardo Thomaz

[1]Centro Universitário FEI
Av. Humberto de Alencar, 3972-B – 09850-901, São Bernardo do Campo, SP

leonardoantunes123@gmail.com, estela.eng@hotmail.com, cet@fei.edu.br

Abstract. *In this work, we extend a standard and successful acoustic feature extraction approach based on trigger selection to examples of Brazilian Bossa-Nova and Heitor Villa Lobos music pieces. Additionally, we propose and implement a computational framework to disclose whether all the acoustic features extracted are statistically relevant, that is, non-redundant. Our experimental results show that not all these well-known features might be necessary for trigger selection, given the multivariate statistical redundancy found, which associated all these acoustic features into 3 clusters with different factor loadings and, consequently, representatives.*

1. Introduction

In the last decade, we have witnessed a significant increase in the number of scientific studies that have identified cognitive differences between musicians and non-musicians [1, 2, 3, 4, 5]. Most of these studies relied on artificial stimuli [4, 5], but recent works on this issue are using naturalistic music pieces, with their corresponding original stimuli, to successfully indicate the cognitive patterns related to human music processing [1, 2, 3].

All these recent works are based on acoustic features extraction from the audio signals to identify the so-called triggers [1], which are instants in the music time series with a rapid increase in the value of a acoustic feature that can be related with cognitive responses. To the best of our knowledge, none of these works has explored such trigger selection approach on Brazilian music.

Therefore, we evaluate whether examples of Brazilian Bossa-Nova and classical music pieces can achieve similar results reported in the aforementioned literature. Moreover, we propose and implement a computational framework based on multivariate statistical analysis to disclose whether all these benchmark acoustic features commonly used in music information retrieval (MIR) tasks are statistically relevant (that is, non-redundant) intending to reduce the number of features required to set triggers for music perception.

2. Methodology

Our methodology can be divided into 3 steps: (I) Music selection; (II) Feature extraction; (III) Feature selection. In the first step we have selected 4 music pieces for the analysis: Garota de Ipanema with two versions, one performed by Tom Jobim (GI1) and the other an instrumental version played by Zimbo Trio (GI2), Bachianas Brasileiras N°2 - O Trenzinho do Caipira composed by Heitor Villa Lobos (BB), and Hungarian Dance N°5 from Johannes Brahms (HD) as a literature and comparative reference. All songs are available on Spotify or IMSLP music libraries and have been gathered with a sampling rate of 44.1 kHz and saved in WAV format.

In the second step, the following 12 benchmark acoustic features utilized to describe audio signals are calculated using the well-known MIRtoolbox (Version 1.71) [1, 6, 7] and Matlab 2015a, decomposing the audio into a 50 milliseconds window overlapped by 50%: (1) Root Mean Square Energy (RMS), (2) Zero Crossing Rate (ZCR), (3) Spectral Rolloff, (4) Spectral Roughness, (5) Brightness, (6) Spectral Entropy, (7) Spectral Flatness, (8) Spectral Skewness, (9) Spectral Kurtosis, (10) Spectral Centroid, (11) Spectral Spread and (12) Spectral Flux. A detailed description of these features can be found in the user manual of the MIRtoolbox [7].

In the third step, we estimate whether all these 12 features, concatenated directly from the previous analysis windows, are statistically relevant through a cluster analysis using Factor Analysis (FA) with *varimax* rotation [8]. The number of factors retained in FA is equal to the number of principal components extracted from data that have eigenvalue greater than 1. We have selected the representative acoustic feature within each cluster as the one with the highest loading on each factor. The K-means algorithm has been applied to identify clusters that describe statistically similar acoustic features.

3. Results

Our results show that for all songs using only three principal components is possible to explain more than 80% of the data variance. Thus, the number of factors used in FA is equal to three. The factor loadings obtained by FA disclose the correlation among the acoustic features and the clusters presented in Figure 1 were obtained with the K-means algorithm.

Clearly, from Figure 1, we can see the same inter-cluster grouping of the acoustic features for all music pieces. Thus, regardless of the musical genre or whether it is an instrumental or vocal music piece, there are the same acoustic feature memberships between the same clusters. Some of these memberships might be expected. For instance, the S. Skewness (acoustic feature 8) and S. Kurtosis (9) are in the same cluster 1 for all four songs, since both provide information about the type and magnitude of audio signals departures from normality. However, it is interesting to notice that the importance of each feature within each cluster is not equal. In other words, for each song (GI1, GI2, BB and HD), there are 3 non-redundant clus-

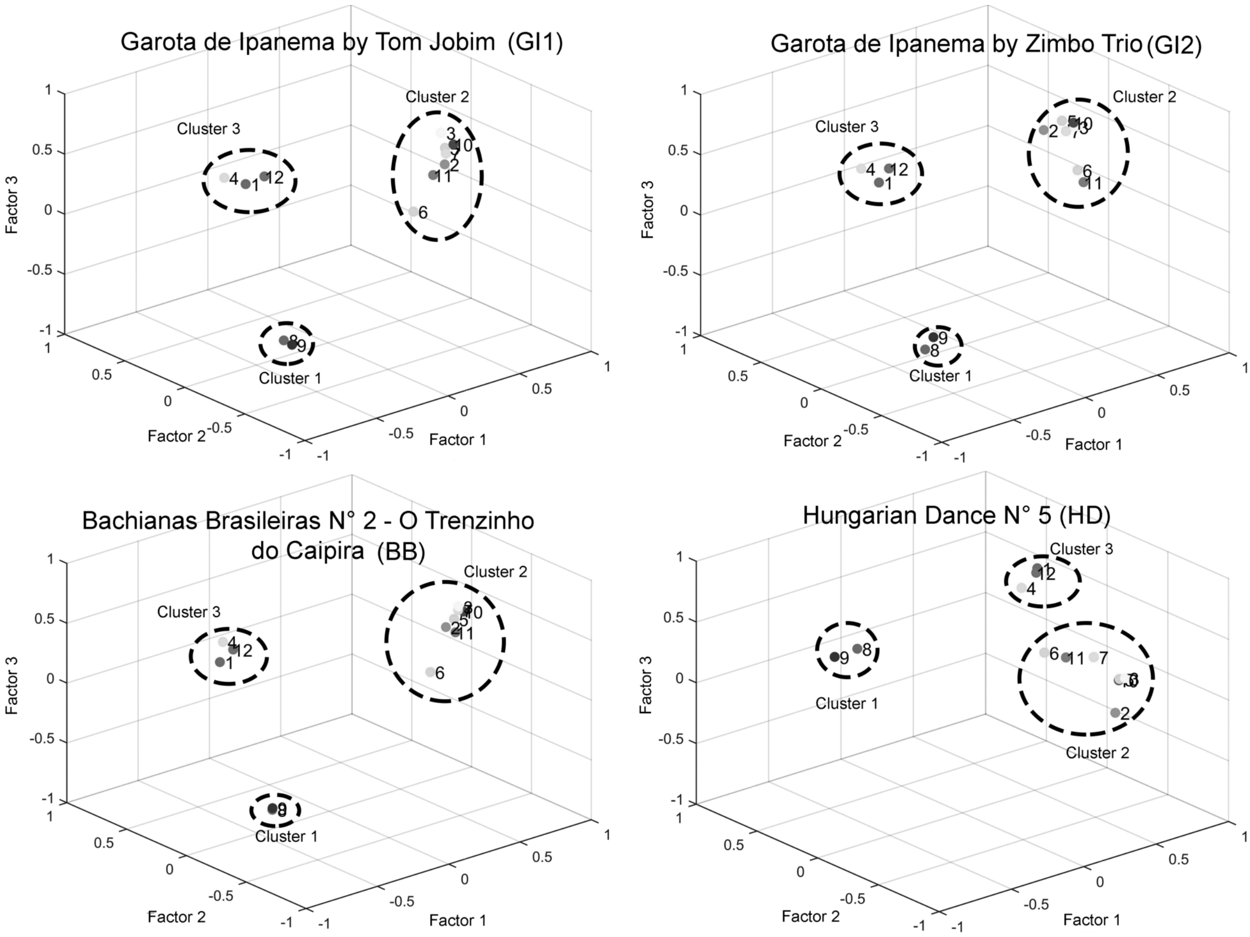


Figure 1: Factor loadings of the acoustic features extracted from the audio signals, being possible to observe the formation of clusters between the following features: Cluster 1 - 8 (S. Skewness) and 9 (S. Kurtosis); Cluster 2 - 2 (ZCR), 3 (S. Rolloff), 5 (Brightness), 6 (S. Entropy), 7 (S. Flatness), 10 (S. Centroid) and 11 (S. Spread); Cluster 3 - 1 (RMS), 4 (S. Roughness) and 12 (S. Flux).

ters, but with distinct acoustic feature representatives, as follows [cluster1, cluster2, cluster3]: GI1 [9, 10, 1]; GI2 [8, 5, 1]; BB [8, 10, 1]; HD [9, 3, 1].

4. Conclusion

In this work, we showed that it is possible to reduce the number of acoustic features required to investigate the cognitive patterns evoked during music listening, given the statistical redundancy found by FA, which grouped all the 12 benchmark acoustic features into only 3 clusters.

This is an exploratory study that indicates a similar clustering behavior between a couple of Brazilian Bossa-Nova and classical music pieces when performing benchmark acoustic features extraction. The same inter-cluster behavior might be achieved in other musical genres, requiring further investigation. However, the distinct intra-cluster acoustic feature loadings highlight the importance of selecting the most relevant features within each cluster to properly representing statistically each music piece for perception analysis.

Acknowledgments

This study was financially supported in part by FEI (PBIC 074/18), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001, and the INCT MACC, process 465586/2014-7. We would like to thank all the volunteers who took part in the experiment.

References

[1] Hanna Poikonen, Vinoo Alluri, Elvira Brattico, Olivier Lartillot, Mari Tervaniemi, and Minna Huotilainen. Event-related brain responses while listening to entire pieces of music. *Neuroscience*, 312:58–73, 2016.

[2] Andjela Markovic, Jürg Kühnis, and Lutz Jäncke. Task context influences brain activation during music listening. *Frontiers in human neuroscience*, 11:342, 2017.

[3] Pasi Saari, Iballa Burunat, Elvira Brattico, and Petri Toiviainen. Decoding musical training from dynamic processing of musical features in the brain. *Scientific Report*, 708(8):1–12, 2018.

[4] Peter Vuust, Elvira Brattico, Miia Sppanen, Risto Naatanen, and Mari Tervaniemi. The sound of music: Differentiating musicians using a fast, musical multi-feature mismatch neg-

ativity paradigm. *Neuropsychologia*, 50(7):1432–1443, June 2012.

[5] C.A. Mikutta, G. Maissen, A. Altorfer, W. Strik, and T. Koenig. Professional musicians listen differently to music. *Neuroscience*, 268:102 – 111, 2014.

[6] Vinoo Alluri, Petri Toiviainen, Iiro P Jääskeläinen, Enrico Glerean, Mikko Sams, and Elvira Brattico. Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm. *Neuroimage*, 59(4):3677–3689, 2012.

[7] Oliver Lartillot. *MIRtoolbox 1.6.1 Users Manual*, 2014.

[8] Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. Pearson, 6th edition, 2007.

The taste of scales and chords

Alan Taitz[1], Diego Shalom[1], Marcos A Trevisan[1] and Bruno Mesz[2]

[1] Dept. of Physics, University of Buenos Aires and Institute of Physics Buenos Aires (IFIBA) CONICET, Pabellón 1 Ciudad Universitaria, 1428EGA Buenos Aires, Argentina, [2] MUNTREF Tecnópolis, Universidad Nacional de Tres de Febrero, Buenos Aires, Argentina.

Abstract

Reliable crossmodal correspondences between basic tastes and music features have been found in recent studies [1,2]. In this work, we explore associations between scales, chords and tastes. Several of these elementary musical structures show non-random patterns of matching with basic tastes. Moreover, their aggregate dyadic consonance [3] anti-correlates with the relative frequency of their matching to bitter taste.

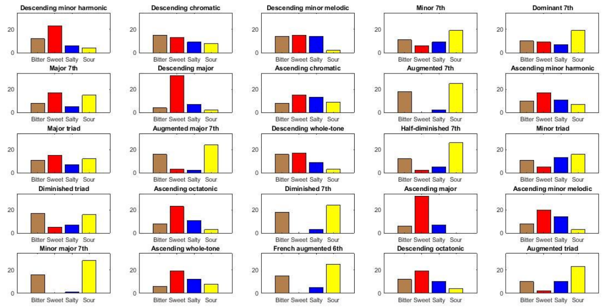


Figure 1 | Frequency of taste matches for each musical stimulus for the 45 participants.

1. Online association experiment

Forty five non-musicians (26 females; age 356 years) filled a Google Form survey. They were presented with 25 short sound files, each containing a scale or a chord, and asked to match a single taste word to each of them, among the four basic taste words bitter, sweet, salty and sour. All scales and chords were presented in a MIDI classical piano timbre, in the middle C4 octave. Audios were mastered to have a homogeneous loudness of where is the duration of the corresponding sound file. Each participant listened the audio files over earphones, adjusting the computer sound volume to a comfortable level with a practice video and then leaving it fixed. The scales and chords were based on C. The duration of chords and scales was fixed to 2s and 0.5s respectively; scales were presented in ascending and descending form. E

Scale types were: *Major, Minor Melodic, Minor Harmonic, Chromatic, Whole-tone, Octatonic.* Chord types were: *Major triad, Minor triad, Diminished triad, Augmented triad, Minor 7th, Major 7th, Minor major 7th, Diminished 7th, Dominant 7th, Half-diminished 7th, Augmented 7th, Augmented major 7th, French augmented 6th.*

2. Results

Associations for each taste are shown in Figure 1. We ran a chi-squared test under the null hypothesis of random responses (25% for each taste). For 13 out of the 25 musical structures the null hypothesis was rejected (significance level of 0.01).

These 13 structures are shown in Figure 2, where we performed a hierarchical clustering of the taste frequency profiles based on Euclidean distance. Two main clusters emerged: scales (mainly sweet or bitter and sweet associations) and chords (mainly sour or bitter and sour).

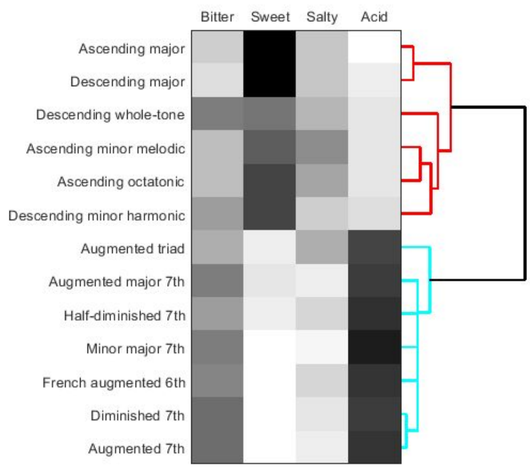


Figure 2 | Proportion of taste matches and clustering for the 13 structures with non-random matches.

3. Aggregate Dyadic consonance and bitter taste

To test the influence of interval content or melodic/ harmonic consonance of the scales and chords on taste choices, we computed aggregate dyadic consonance [3] for each structure with non-random taste matchings. This measure is based on the interval vector obtained from the collection of pitches in the structure, and consists of a weighted average of the interval vector, where the weights of each interval class derive from empirical studies of consonance perception.

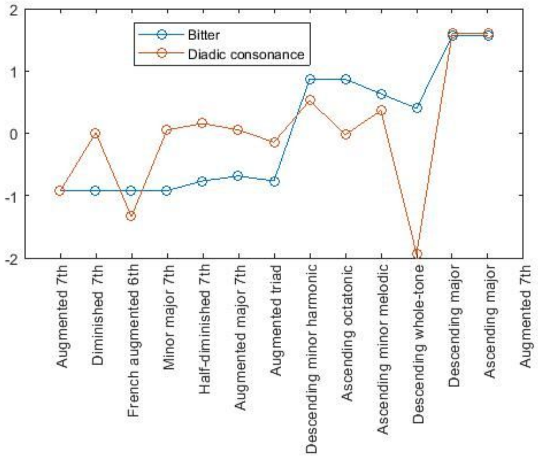


Figure 3 | Z-scores of Bitter matches and Aggregate Dyadic Consonance.

We computed Pearson correlation of this measure with frequency of responses for each taste, over all the 25 musical structures. There is a weak anticorrelation ($r=-0.4$, $p=0.046$) in the case of bitter taste, and no significant correlation for the other tastes. The Z-scores of consonance and bitter responses are plotted in Figure 3.

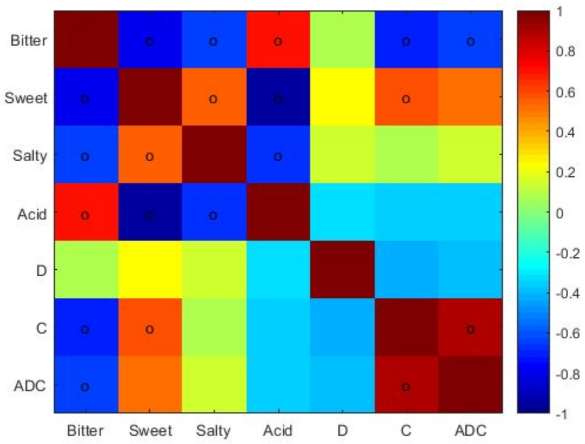


Figure 4 | Correlations between frequencies of taste matches and measures of consonance/dissonance: D (Interval Dissonance Rate), C (Correlation consonance), ADR (Aggregate Dyadic Consonance). Circles indicate significant correlations (significance level of 0.05).

More generally, in Figure 4 we present correlations between the associated taste frequencies and three measures of consonance/dissonance: Interval Dissonance Rate [4], Correlation consonance [3] and Aggregate dyadic consonance. In agreement with previous results [1-2], we find positive correlations between consonance and sweetness, and a strong anticorrelation between sour and sweet taste-music associations.

4. Discussion

Previous work on crossmodal correspondences has focused either on single notes or complex music. Here we considered the intermediate domain of basic music scales and chords, keeping constant the duration, intensity, register and timbre of the audio files. We show that the appearance of correspondences still occurs in this case, and that scales differ from chords in their matched taste profiles. In agreement with previous results [3], bitter and sweet present a negative and positive correlation respectively with consonance. However, dependence of sour on dissonance reported in other studies [2] were absent, suggesting that they may be context-dependent.

In the near future we plan to extend this study to consider a greater variety of musical structures and also include timbre as a variable parameter in the stimuli.

References

[1] Mesz, B., Trevisan, M. A., & Sigman, M. (2011). The taste of music. *Perception*, *40*(2), 209-219.

[2] Knöferle, K., & Spence, C. (2012). Crossmodal correspondences between sounds and tastes. *Psychonomic bulletin & review*, 1-15.

[3] Huron, D. (1994). Interval-class content in equally tempered pitch-class sets: Common scales exhibit optimum tonal consonance. *Music Perception: An Interdisciplinary Journal*, *11*(3), 289-305.

[4] Mamedov, N., & Peck, R. (2017). The Interval Dissonance Rate: An Analytical Look into Chromaticism of Chopin's Op. 10 No. 2 and Scriabin's Op. 11 No. 2. In *Bridges 2017 Conference Proceedings* (pp. 475-478). Tessellations Publishing.

Automatic onset detection using convolutional neural networks

Willy Garabini Cornelissen[1] **, Mauricio Alves Loureiro**[1]

[1]CEGeME – Escola de Musica/UFMG
Av. Antonio Carlos, 6627, Pampulha - Belo Horizonte - MG

willy@ufmg.br, mauricio@musica.ufmg.br

Abstract. *A very significant task for music research is to estimate instants when meaningful events begin (onset) and when they end (offset). Onset detection is widely applied in many fields: electrocardiograms, seismographic data, stock market results and many Music Information Research(MIR) tasks, such as Automatic Music Transcription, Rhythm Detection, Speech Recognition, etc. Automatic Onset Detection(AOD) received, recently, a huge contribution coming from Artificial Intelligence (AI) methods, mainly Machine Learning and Deep Learning. In this work, the use of Convolutional Neural Networks (CNN) is explored by adapting its original architecture in order to apply the approach to automatic onset detection on audio musical signals. We used a CNN network for onset detection on a very general dataset, well acknowledged by the MIR community, and examined the accuracy of the method by comparison to ground truth data published by the dataset. The results are promising and outperform another methods of musical onset detection.*

1 Introduction

The extraction of onset times from a spectrogram is equivalent of detecting edges on an image. Oriented edges in images can be found by convolution with small filter kernels even of random values. This lead to the idea of training a Convolutional Neural Network (CNN) to find onsets in spectrogram excerpts. Convolutional learning in Music Information Research has been applied before for genre and artist classification [[1], [2]]. Their application on onset detection, a comparably low-level task, achieve promising results. [[3],[4]]

2 Onset definition

Flights and Rach [5] defined the perceptual beginning of a musical sound as a time instant in which the stimulus is perceived for the first time. The physical onset, however, can be defined as the instant at which the generation of the stimulus was initiated. Usually, the perceptive onset is delayed in relation to physical onset. The time interval between the physical and the perceptual initiation results, among other things, from the fact that most musical and speech stimuli do not begin at levels near their maximum, but begin with gradually increasing amplitudes. At the beginning of the physical stimulus, the amplitude level is often too low to attract the conscious attention of the listener. In this work we will follow the definition of onset proposed by Bello: initial instant of a sound event [6].

3 Convolutional Neural Networks

Convolution is the process of adding each element of the image to its local neighbors, weighted by the kernel. The

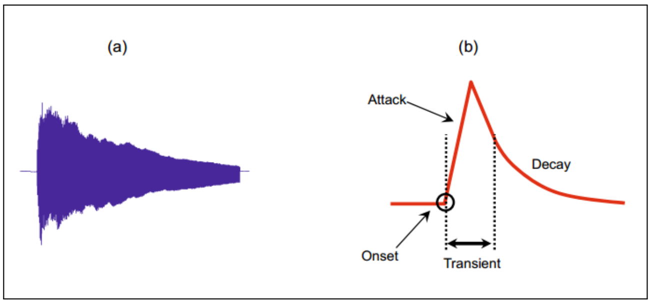


Figure 1: A note played on the piano (a) and its amplitude envelope indicating the regions of attack, transient, onset and decay (b) adapted from [7, p. 305]

.

kernel, or convolutional matrix, is multiplied to the original matrix, resulting in a single value, as shown in the example. This operation gives to CNN a high accuracy on image recognition tasks, although the computational cost is high and need a lot of training data.

3.1 Computer Vision and Machine Listening

CNNs are great for computer vision task, bu to apply it in spectrograms for machine listening, some challenges must be overcomed:

- **Sound objects are transparent:** visual objects and sound events in a image behaves differently. The problem is that discrete sound events do not separate into layers on a spectrogram: Instead, they all sum together into a distinct whole. Visual objects are "individualized", and sound events in a spectrogram cannot be assumed to belong to a single sound, as the "magnitude of of that frequency could have been produced by any number of accumulated sounds or even by the complex interactions between sound waves such as phase cancellation. This makes it difficult to separate simultaneous sounds in spectrogram representations." [8]
- **Meaning of axes:** one of the big advantages of a CNN is that they are built on the assumption that features of an image carry the same meaning regardless of their location. But dealing with spectrograms, the two dimensions represent fundamentally different units, one being strength of frequency and the other being time. Moving a sound event horizontally is just shift in time, but to move it vertically causes a notable change on its nature. Therefore, the spatial invariance that

2D CNNs provide might not perform as well for this form of data.

- **Sounds are not local:** the frequencies represented in a spectrogram are not locally grouped. Instead of this, they move together according to a common relationship(the fundamental frequency).
- **Sound is a temporal event:** in a visual scenario, objects persist on time and can be re-scanned. This it not true true for sound events. This is why it makes sense to refer to these phenomena as sound events rather than sound objects.

4 Methodology

4.1 Data

Sebastian Bock, the author of the model called state of the art (SOTA) in onset detection [4], prepared a dataset that we used as graund-truth to illustrate onset detection using CNN. The dataset contains 321 audio excerpts taken from various sources. 87 tracks were taken from the dataset used in [9], 23 from [6], and 92 from [10].

4.2 Detection Method

The method use two convolutional and pooling layers to perform local processing, a single fully-connected hidden layer and a single output unit.

4.3 Evaluation Metric

For the comparison of detected onset with the ground-truth, if the detected instant falls within a tolerance time-window around that instant, it is considered as a true positive (TP). If not, there is a false negative (FN). The detections outside all the tolerance windows are counted as false positives (FP). Doubled onsets (two detections for one ground-truth onset) and merged onsets (one detection for two ground-truth onsets) will be taken into account in the evaluation. Doubled onsets are a subset of the FP onsets, and merged onsets a subset of FN onsets. **Precision, Recall** and **F-measure** were used to evaluate the performance of the detection.

5 Results

Figure 2 illustrate the power of the state of the art algorithm. The histogram of this figure shows that most outcomes of f-measure lies between 0.8 and 0.9 with median 0.9 and a and a concentration of files that achieve a value between 0.75 and 0.95 for all these metrics.

6 Conclusion

This work showed how machine learning with convolutional neural networks was well integrated in the process of detecting onsets and has been showing important contributions to the optimization of more traditional methods. It was verified that this method performs very well in a large and generic dataset, confirming the state of the art achieved by CNN on AOD.

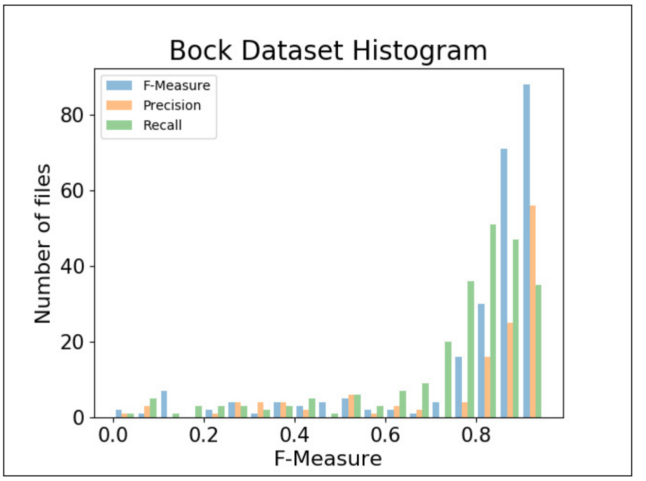


Figure 2: Bock Dataset Histogram with CNN Onset Detection. Blue for *f-measure*, orange for *precision*— and green for *recall*.

References

[1] Tom LH Li, Antoni B Chan, and A Chun. Automatic musical pattern feature extraction using convolutional neural network. In *Proc. Int. Conf. Data Mining and Applications*, volume 161. Citeseer, 2010.

[2] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based music classification with a pre-trained convolutional network. In *12th International Society for Music Information Retrieval Conference (ISMIR-2011)*, pages 669–674. University of Miami, 2011.

[3] Jan Schlüter and Sebastian Böck. Musical onset detection with convolutional neural networks. In *6th international workshop on machine learning and music (MML), Prague, Czech Republic*, 2013.

[4] Jan Schlüter and Sebastian Böck. Improved musical onset detection with convolutional neural networks. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 6979–6983. IEEE, 2014.

[5] Joos Vos and Rudolf Rasch. The perceptual onset of musical tones. *Perception & psychophysics*, 29(4):323–335, 1981.

[6] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on speech and audio processing*, 13(5):1035–1047, 2005.

[7] Meinard Müller. *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer, 2015.

[8] What's wrong with spectrograms and cnns for audio processing? `https://t.co/qequ0e3l18`. (Accessed on 05/14/2019).

[9] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves. Universal onset detection with bidirectional long-short term memory neural networks. In *Proc. 11th Intern. Soc. for Music Information Retrieval Conference, ISMIR, Utrecht, The Netherlands*, pages 589–594, 2010.

[10] André Holzapfel, Yannis Stylianou, Ali C Gedik, and Barış Bozkurt. Three dimensions of pitched instrument onset detection. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1517–1527, 2010.

Harmonia: a MuseScore's plugin to teach music

João Teixeira Araújo, Rogério Tavares Constante,
Flávio Luiz Schiavoni, Octávio Deluchi

[1] Arts Lab in Interfaces, Computers, and Everything Else - ALICE
Federal University of São João del-Rei - UFSJ
São João del-Rei - MG

teixeira.araujo@gmail.com, rogerio_constante@hotmail.com

fls@ufsj.edu.br, ocdeluchi@gmail.com

Abstract. *Information and Communication Technologies (ICTs) have been characterized as a very effective resource for promoting innovation in the way of teaching and learning. In relation to the musical area, computer software of musical notation, like MuseScore, has been more and more used for the musical compositions and to teach and learn music writing, musical arrangement, composition and counterpoint. MuseScore is a free software that can easily be applied to academic purposes, such as universities, for teaching students in music fields and can also be used in the professional life of students who have graduated from courses that use it. In addition, it allows the implementation of plugins for various purposes, such as the analysis of scores in relation to various preset parameters. In this context, this work aims to present the development of the Harmonia, an open source plugin for MuseScore focused on teaching musical analysis and automatic verification of scores based in harmony predefined rules.*

1 Introduction

Currently, we are in the era in which technological evolution is always present in our daily lives, which has brought changes in several areas, whether they are aimed at teaching or for the development of new tools. In general, the ways of teaching and learning a particular content have changed considerably over the time. The computer, for example, has been increasingly used, taking into account its ability to provide various functionalities in the form of implementation of computer programs intended for a multitude of purposes [1].

As digital technologies have become ubiquitous tools, being present in the daily lives of people in a very expressive way, it becomes very promising the use of these resources in the area of education. However, it is necessary to know how to manipulate such resources and use them strategically, with a focus on knowledge production, in a reflexive and creative way [2].

Among the areas that are promising for the use of new technologies is the education, and in particular, the music education. The new technologies have challenged musical education to undergo a series of transformations, challenging teachers to modify their respective educational concepts, didactic perspectives and to lead them to think about the new demands and possibilities regarding interactions with students [3].

This work presents the Harmonia plugin, an extension of MuseScore focused on teaching and practicing music that aims to support the musician or student of music in the automatic analysis and verification of musical scores based on aspects frequently present in the study of harmony.

2 Related tools on musical education

Our work began with a survey of data on the educational tools aimed at the study of harmony until then. We focus on those software that perform the task of analyzing and investigating, from a score, the following aspects: voice leading; treatment of dissonances; melodic construction; texture and chord structure.

We used the following criteria to define what would be desirable in a tool to help automatic harmonic review. In the category **voice leading**, we chose the elements of *8th and 5th parallel*, *other parallel intervals*, *all voices in the same direction*, *hidden 8th and 5th* , *overlapping*, *cross relation*, *unison*, *the tendency tones*. In **dissonance treatment** are the standards of *suspension* and *passing note*, as well as the treatment of the bass, in the 2nd inversion chord, among others. In **melodic construction**, we select the aspects *augmented and diminished intervals*, *melodic leaps*, *definition of melodic contour through focal point*, *melodic variety* and the *extension of the melodies*. In the **texture** category there are the issues of the *spacing the voices* and the *vocal range*. In **chord structure**, there are aspects of *folding* and *omission of notes*.

A first possibility that we considered to find musical tools to help students with harmony tasks was to find it in Moodle plugin database. Moodle[4] is a widely used learning management system that is extensible by the use of plugins. Searching for moodle plugins with the term "harmony" there were no plugins intended for this purpose. A search for the term "music" in the Moodle database led us to 4 plugins which did not fit our selection criteria.

Another possibility of tools could be to integrate musical analysis in a score notation tools. Since we intended to create an open source tool, we looked for an open source notation tool to base our research. We chose MuseScore, an open source score editor that also can be extended by plugins. In the MuseScore plugins, we found more than 70 plugins, but only two of them fitted our search criteria, highlighting the *Check Harmony Rules* plugin that verifies 3 aspects of the voice leading and 2 aspects of the melodic construction.

We also verified 8 musical software, but only the Pizzicatto Composition Pro fitted the criteria defined by allowing the verification of 4 aspects of the voice leading, 2 aspects of the melodic construction, 1 aspect of the treatment of dissonances and 2 aspects of the texture.

3 Harmonia plugin

Due the lack of tools to meet this field of music teaching, we decided to create a plugin for MuseScore that would allow musical analysis based on the presented criteria. The pedagogical approach of the various aspects of the study of harmony is variable, according to the different authors, (like [5, 6], and others) and with the various teaching-learning proposals. A tool that performs verification based on fixed parameters in a given standard can be useful in a limited field of pedagogical approaches.

Our MuseScore plugin, called Harmonia, performs several checks of relevant musical aspects to the study of part writing exercises written in MuseScore, such as: voice leading, dissonance treatment, melodic construction, texture and chord structure. The configuration of the parameters of analysis and verification of the part writing exercises can be done directly in the plugin interface and recorded in a file for future use, in the various contexts.

Figure 1 demonstrates the Harmonia interface, divided into three main areas: the upper area, aimed at selecting which type of verification to execute, that could be under voice leading, dissonance treatment, melody, texture or chords; the center / left area, for selecting which parameters to verify under the selected verification type, such as dissonance use which are not a suspension pattern; and the area on the right, focused on selecting global parameters, such as the key to be checked.

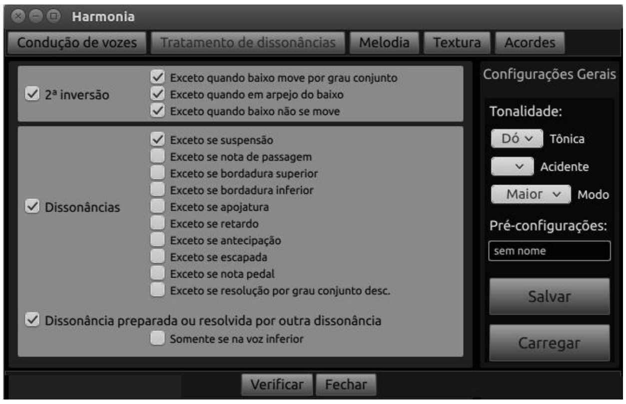


Figure 1: Example of the Harmonia's interface used to set up the tool

After the parsing process, the plugin lists a sequence of checked clickable items, indicating the type of scan and where the issues occurs (bars and voices). When clicked, the item is selected and the notes related to the respective problem become red. If desired, the user can continue clicking on the following items in the list to view them in the score. As the changes to the score are made to correct the issues detected, the list can be updated on the fly.

4 Final considerations

It is frequent in the teaching-learning process of harmony to perform a great amount of exercises, among which, those of part writing. The computational automation of the verification process of a great number of aspects, in the correction of exercises, leaves this activity more agile and fast. A student of music can have greater autonomy to seek solutions to issues detected by an automatic tool almost immediately, without having to wait for the teacher's analysis. Of course, it does not mean that the teacher will be expendable. The software verifies only a portion of the issues dealt with in the study of harmony, leaving aside so many aspects that are not objectively verifiable.

In this paper, we presented a tool to help students and teachers to automatically correct harmony phrases trying to give feedback on the fly for those who wants some help in musical writing like composition, arrangement and counterpoint. The presented plugin is moldable to meet the various pedagogical proposals, through the configuration of what is intended to be checked. This feature allows it to be used as a support tool in conjunction with different methods and reference textbooks of harmony.

We chose to write it as a plugin for the MuseScore since our purpose is consistent with the concept of FLOSS, especially considering its applicability in the academic field. Because it is a free software, MuseScore is a tool that can easily be applied in universities, since its cost is zero and from it, the student can learn to use it and consequently use it even in the future, in their professional life, without having to worry about copyright and other factors that the tools paid end up implying. In this way, the development of plugins for MuseScore, ends up being quite promising, since from them it is possible to implement several functionalities that sometimes a free tool does not have.

5 Acknowledgments

Authors would like to thanks to all ALICE members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq and FAPEMIG.

References

[1] Ana Cristina Vieira Gouveia. Educação musical e as novas tecnologias. ferramentas de apoio ao docente de música. 2015.

[2] Luciano Luan Gomes Paiva. Tecnologias digitais na educação musical: discussões emergentes omunicação. *XI Conferência Regional Latino-Americana de Educação Musical da ISME*, 2017.

[3] Susana Ester Krüger. Educação musical apoiada pelas novas tecnologias de informação e comunicação (tic): pesquisas, práticas e formação de docentes. *Revista da ABEM*, 14(14), 2014.

[4] Martin Dougiamas and Peter Taylor. Moodle: Using learning communities to create an open source course management system. In *EdMedia+ Innovate Learning*, pages 171–178. Association for the Advancement of Computing in Education (AACE), 2003.

[5] Arnold Schoenberg. *Harmonia*. Editora UNESP, São Paulo, SP, 2001.

[6] Stefan Kostka and Dorothy Payne. *Tonal Harmony*. Alfred A. Knopf, New York, 1984.

Audio Encryption Scheme based on Pseudo-orbit of Chaotic Map

Magalhães, E.P.[1] **, Santos, T.A.**[1] **, Nepomuceno, E.G.**[1*]

[1]Control and Modelling Group (GCOM)
Department of Electrical Engineering
Federal University of São João del-Rei
Pça. Frei Orlando, 170 - Centro - 36307-352 - São João del-Rei, Minas Gerais, Brasil

`eduardopintomagalhaes@gmail.com, contato@tsantos.com.br, nepomuceno@ufsj.edu.br`

Abstract. *Chaos-based encryption uses a chaotic dynamic system to encrypt a file. The aim of this study was to investigate the use of the chaotic Cubic Map to encrypt data, in particular, audio files. A simple algorithm was developed to encrypt and decrypt an audio data. The effectiveness of the method was measured by means of the correlation coefficient calculation, spectral entropy and also by comparing waveforms. The measurements were shown to lead to satisfactory confusion levels of the original data, within a few seconds. This indicates that the Cubic Map can be used as a source for encryption keys, with as good or better security indicators when compared to other schemes.*

1 Introduction

Chaos-based encryption has received much attention since the work of Matthews [8], and has been persistently studied ever since. An encrypted data is obtained from the logical exclusive or operation between some data and a chaotic pseudo-orbit. Chaos-based encryption has many possible uses in the digital security field and different chaotic systems have also been investigated as a potential improvement to security and performance [4, 13, 2, 14, 6, 3].

Several works regarding cryptography using dynamic chaotic systems were proposed in the literature. For example, Sheu [12] proposed a speech encryption algorithm using fractional chaotic systems and Kordov and Bonchev [5] proposed an algorithm of audio-based encryption using a circular map. However, to the best of the authors knowledge and bibliographic review, no application of the Cubic Map was developed for this purpose.

This research aims to apply the Cubic Map to a very simple encryption algorithm and measure its performance by comparing the results with more complex schemes already published. This paper is organized as follows: Section 2 deals with the preliminary concepts; Section 3 discusses the proposed methodology for cryptography and statistical analysis; Section 4 presents the results, as well as their analyzes and finally in Section 5, the conclusion of this work.

*This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, the Ministry of Science, Technology, Innovation and Communications (MCTI) and the Federal University of São João del Rei.

2 Preliminary Concepts

In this section, the theoretical concepts needed to carry out this work are presented.

2.1 Chaotic dynamic systems

Chaotic dynamic systems have been studied from the work of [7]. The accepted definition of chaos found in the literature can be properly explained by Banks et al.

Definition 1: *[1]. Let $f : X \to X$ be a chaotic system. This system is chaotic when it has the three following properties, such as being transitive, dense in X and sensitive to the initial conditions: f is transitive; the periodic orbits of f are dense in X and f is sensitive to the initial conditions.*

2.2 Cubic Map

The Cubic Map is a map that has a chaotic behavior from the value r, known as the bifurcation parameter. This map is a discrete and dynamic system described by the equation below:

$$f_r(x) = rx^3 + (1 - r)x \tag{1}$$

3 Methodology

3.1 Audio Encryption Process

The current study required simulating and analyzing audio file encryption and decryption as if in a real-world file exchange. Countdown for a space launch was chosen as subject as it has subtle and strong features. The audio was obtained through NASA's website and refers to the liftoff of Atlas V from Cape Canaveral [1]. One interval extensions of the Cubic Map was simulated using a set of conditions and parameters known to be chaotic, and care was taken to analyze chaotic indicators [10, 9].

The pseudo-orbit was simulated using $r = 3.6$ and 1900 initial conditions varying linearly from -1 to 1, on MATLAB R2018a running on a GNU Linux machine with Intel(R) Core(TM) i7-7700HQ CPU @

[1]All the audio files referenced in this work can be acessed through: `https://bit.ly/2J0lbEF`.

2.80GHz. Having more initial conditions means reducing the length of each simulation, yielding more precise results [10].

By obtaining the same numerical type, the encryption process could begin. This consists of applying logical Exclusive Or ($\oplus$) operation bit by bit. So, the encrypted audio is obtained by Equation 2.

$$EncryptedAudio = \sigma_{Norm} \oplus ConvertedAudio \quad (2)$$

Encrypted data was then statistically analyzed and searched for vulnerabilities and similarities with the original data.

The encryption process wouldn't be successful if the data could not be recovered. Thus, by simulating the same pseudo-orbit and applying Exclusive Or operation again, as in Equation 3 the data was restored to its original state.

$$ConvertedAudio = \sigma_{Norm} \oplus ConvertedAudio \quad (3)$$

In order for the file to be readable, the inverse process of audio conversion must be made, converting it back to double precision, subtracting 32768 and converting back to 16 bit signed integer.

3.2 Spectral Entropy

The spectral entropy of the signal was defined as the energy distribution as a function of frequency. Therefore, the frequency spectrum was calculated by FFT (Fast Fourier Transform). However, large FFT sizes can produce high-frequency resolutions, while low FFT sizes have the opposite effect. In order to achieve a high resolution in the frequency axis, the overlay percentage feature was used. This feature allows the frequency axis to be stretched on the spectrogram graph by processing parts of the frequency series. The resolution was then set in a window size of 2048 points and a percentage of overlap in 75%. The spectral spectrum is normalized between 0 and 1. Thus, the spectral entropy was normalized between 0 (total regularity) and 1 (maximum irregularity).

4 Results

4.1 Waveform Analysis

The waveform plots can be seen in Figure 1. The graph represents the audio amplitude distributed in time. It can be observed that the original data in Figure 1 (a) loses its features and cannot be distinguished in Figure 1 (b). By means of the Figure 1 (b) it is observed that the encrypted signal remains secure under the data transmission process, since the keyspace that is based on the initial conditions of the Cubic Map and the bifurcation parameter r was determined to be large enough.

The Figures 2 (a) and (b) illustrate the main band occupied by the signals. The frequency of the signal occupying 99% of its band is 5.148 kHz, while the frequency of the cryptographic signal occupying 99% of its band is 21.894 kHz.

5 Conclusion

A new scheme for chaos-based encryption was developed. The XOR encryption scheme together with the Cubic Map is investigated analytically and compared with other results on the literature [5, 11]. The Cubic Map proved to be robust under the aspect of the secure key from its initial conditions since it was possible to satisfactorily describe the complexity of the regularity of the original and cryptographic signal.

References

[1] John Banks, Jeffrey Brooks, Grant Cairns, Gary Davis, and Peter Stacey. On devaney's definition of chaos. *The American mathematical monthly*, 99(4):332–334, 1992.

[2] X Chai, Y Chen, and L Broyde. A novel chaos-based image encryption algorithm using dna sequence operations. *Optics and Lasers in engineering*, 2017.

[3] R Guesmi, MAB Farah, A Kachouri, and M Samet. A novel chaos-based image encryption using dna sequence operation and secure hash algorithm sha-2. *Nonlinear Dynamics*, 2016.

[4] Z Hua and Y Zhou. Image encryption using 2d logistic-adjusted-sine map. *Information Sciences*, 2016.

[5] Krasimir Kordov and Lachezar Bonchev. Using circle map for audio encryption algorithm. *Mathematical and Software Engineering*, 3(2):183–189, 2017.

[6] W Liu, K Sun, and C Zhu. A fast image encryption algorithm based on chaotic map. *Optics and Lasers in Engineering*, 2016.

[7] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2): 130–141, 1963.

[8] Robert Matthews. On the derivation of a "chaotic" encryption algorithm. *Cryptologia*, 13 (1):29–42, January 1989.

[9] Eduardo M. A. M. Mendes and Erivelton G. Nepomuceno. A very simple method to calculate the (positive) largest lyapunov exponent using interval extensions. *International Journal of Bifurcation and Chaos*, 26(13):1650226, December 2016.

[10] E.G. Nepomuceno, S.A.M. Martins, G.F.V. Amaral, and R. Riveret. On the lower bound error for discrete maps using associative property. *Systems Science & Control Engineering*, 5(1):462–473, January 2017.

[11] Animesh Roy and A. P. Misra. Audio signal encryption using chaotic hénon map and lifting

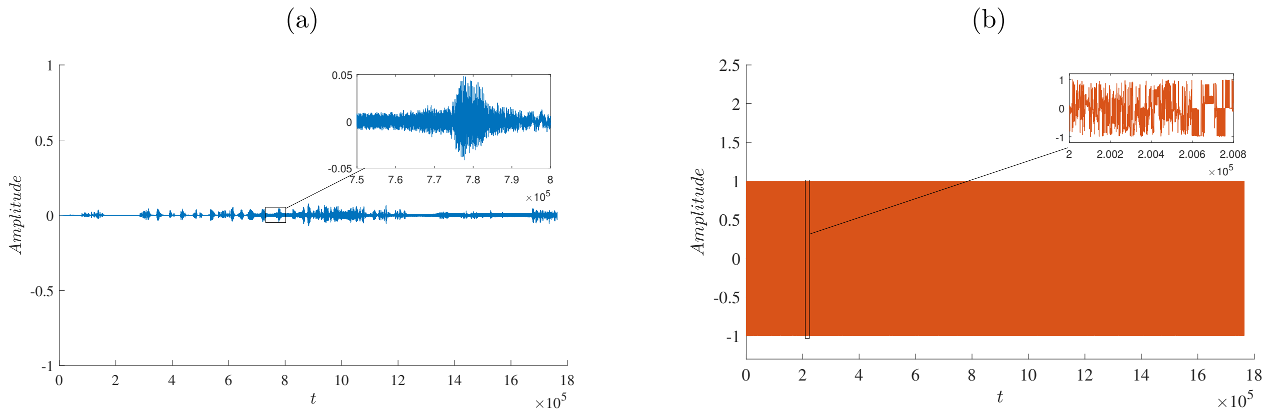


Figure 1: Wave forms (amplitude vs time) [(a) and (b)]. It should be noted that (a)'s amplitude is considerably lower than (b). Also, one should observe that, due to the encryption process, the original data (a) cannot be recognized in (b), as it lost all of its features.

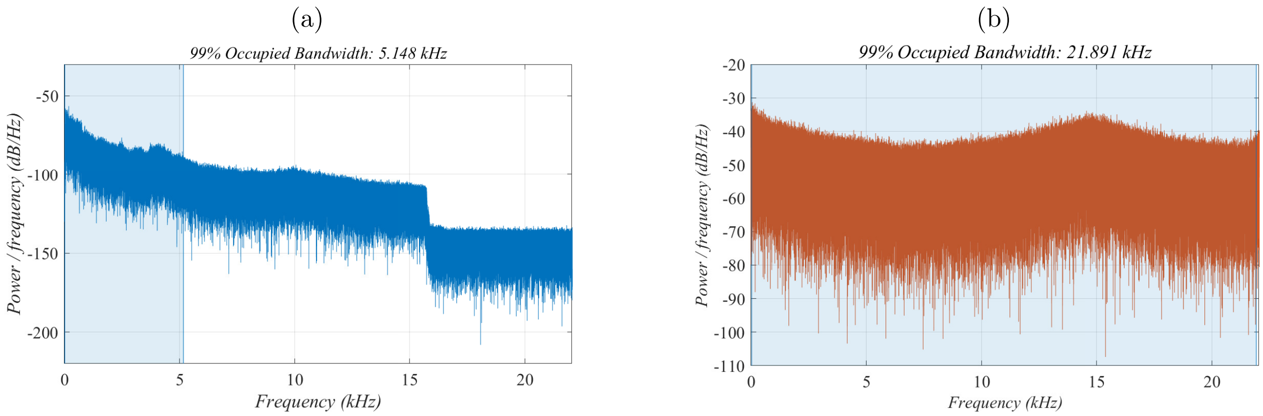


Figure 2: Occupied bandwidth (power/frequency vs frequency) [(c) and (d)] of the original and encrypted audio signals. As can be seen, the signal after the encryption becomes an audio with noise.

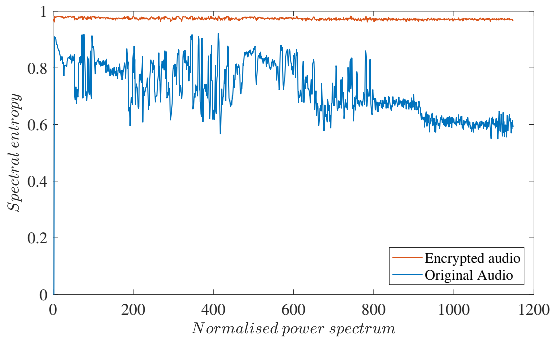


Figure 3: Spectral entropy of the original (blue) and encrypted (brown) audio signals with the following spectral values: between 0,809 and 0,591 for the encrypted signal and 0,9736 (mean value) for the original signal (close to 1). Entropy shows high homogeneous leves of disorder on the encrypted audio, in contrast with the original one.

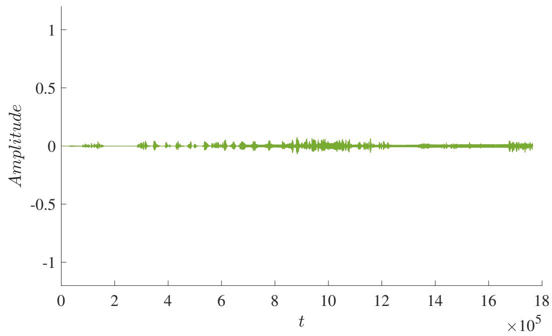


Figure 4: Waveform (amplitude vs time) of the decrypted audio. As can be seen, the audio has been completely decrypted without any loss of information.

wavelet transforms. 2017. doi: 10.1140/epjp/i2017-11808-x.

[12] Long Jye Sheu. A speech encryption using fractional chaotic systems. *Nonlinear dynamics*, 65 (1-2):103–108, 2011.

[13] L Xu, Z Li, J Li, and W Hua. A novel bit-level image encryption algorithm based on chaotic maps. *Optics and Lasers in Engineering*, 2016.

[14] N Zhou, S Pan, S Cheng, and Z Zhou. Image compression–encryption scheme based on hyper-chaotic system and 2d compressive sensing. *Optics & Laser Technology*, 2016.

Digital Design of Audio Signal Processing Using Time Delay

Mauricio Perez[1], Rodolfo Coelho de Souza[1,2*], Regis Alves Rossi Faria[1,3*]

[1] Escola de Comunicações e Artes, Universidade de São Paulo
Av. Prof. Lúcio Martins Rodrigues, 443 – 05508-020, São Paulo, SP

[2] Departamento de Música da FFCLRP, Universidade de São Paulo
Rua Maria M. C. Teles, s/n – 14040-900, Ribeirão Preto, SP

[3] Escola de Artes, Ciências e Humanidades, Universidade de São Paulo
Rua Arlindo Bettio, 1000 – 03828-000, São Paulo, SP

mperez@usp.br, rcoelho@usp.br, regis@usp.br

Abstract

This poster describes the design in PureData of some audio signals processes in real time like *delay*, *echo*, *reverb*, *chorus*, *flanger* e *phaser*. We analyze the technical characteristics of each process and the psychoacoustic effects produced by them in human perception and audio applications. A deeper comprehension of the consequences of sound processes based on delay lines helps the decision-making in professional audio applications such as the audio recording, mixing, besides music composition that employs sound effects in pre-processed or real-time.

1. Introduction

The technique of time delay is simple and versatile. It is often used in audio signal processing for fixing a large set of technical problems, e.g. problems of sound diffusion in concert halls, or it is applied to audio effects that expand the capabilities of acoustic instruments, modifying and creating new timbres for the purpose of music composition.

In this poster, we chose to explore this second trend. Notice that, from the psychoacoustic standpoint, effects based of time delay are related to how the human hearing apparatus receives and interprets the delayed signals. We may understand them as repetitions, some feature of the acoustic space or as the timbre that results from transformations in the spectral dominium.

The first uses in music of sound effects based on time delay goes back to the 1940's. They were delay effects and short echoes that used tape loops in magnetic sound recorders. The procedure was called *tape delays*. The amount of the time delay was ruled by the distance between the reading and recording heads of the devices. This loop arrangement might generate an echo effect that could apply one or many repetitions of the signal to be added to the original signal on another recording device. Until the decade of 1970's this was the basic configuration for an echo system.

As only the digital domain concerns us here, time delay effects can be implemented using a function called *digital delay line*. According to Roads (1996, p.433), this delay type consists in "a data structure called a *circular queue*" in which a list of memory locations, disposed sequentially in the computer's memory, stores the numerical representation of audio samples.

2. Implementation of delay lines in PureData

The design of delay lines using the software PureData (Pd) (Puckette, 2006) can be implemented using the objects [delwrite~] and [delread~]. The first object is responsible for creating the circular buffer, as cited above, containing the audio samples, whereas the second object reads and reproduces them. Both objects receive two arguments on the right side. The first argument of [delwrite~] is the location that stores the buffer and the second stores the time of the buffer.

Roads (1996) remind us about the difference between two types of delay lines: those that use a fixed time of delay and those that use a variable time of delay. The difference is that unities of fixed delay time do not change their time of delay while they process the sound. However, in a unity of variable time of delay, this time can be changed at any moment by varying the reading pointers at each sample period. These two types of delay are also inherent to specific temporal processes. The first case, of unities of fixed time of delay, we can found in the most common processes of delay as delay proper, echo and reverb when generated by delay lines. The second case is used in processes like *chorus*, *flanger* and *phaser*. Therefore, we will start demonstrating the implementation that uses lines of fixed delay time, followed by those that use variable delay lines.

3. Efects with fixed delay lines: *delay, echo* and *reverb*

According to Roads (1996, p.435) fixed delay lines can be arranged in three categories with specific interval times which correspond to three categories of perceptual effects to the human hearing. These three interval times are: short interval times (up to 10 ms), medium interval times (from 10 ms to 50 ms) and large interval times (larger than 50 ms). Short interval times are perceived mainly in the frequency domain as artifacts added to the

* Supported by CNPq

original signal. When a delay line operates between 0,1 and 10 ms, it generates a *comb filter* effect that can reinforce frequencies of the original signal. Medium interval times are perceived as an ambience created around the original sound. This means that the signal is amplified by the sum of the original signal with other signals generated by the delay line. Therefore, the loudness of the original signal seems to increase. Finally, delay lines with large delay times generate a perception of sequential repetitions of the original signal. They correspond to larger spaces with more and distinct reflections.

We will deal now with the design of many delay lines to create the effects of *echo* and *reverb*. These effects can be created with an algorithm that generates a mechanism of feedback of the original signal into the unity of delay processing.

This way, with specific combinations between delay times and feedback gains we may implement different time processes based on fixed delay lines, as those described above, and as shown on Table 1.

Effect	*delay time*	*feedback gain*
comb filter	1-10 ms	0,9
loudness boost	10 - 50 ms	0,5 - 0,3
short echo	50 ms	0,7
large echo	100 ms	0,7 - 0,95
short reverb	100 ms	0,3
large reverb	150 ms	0,5

Table 1: Effects based on delay time and feedback gain

The sonograms of Figures 1A to 1C, produced with the software Spek with audio samples in '.wav' format, mono/44100/32 bits, represent the spectral analysis of different processes applied to a sound sample according to the parameters of Table 1. They allow us to visualize the differences between these processes.

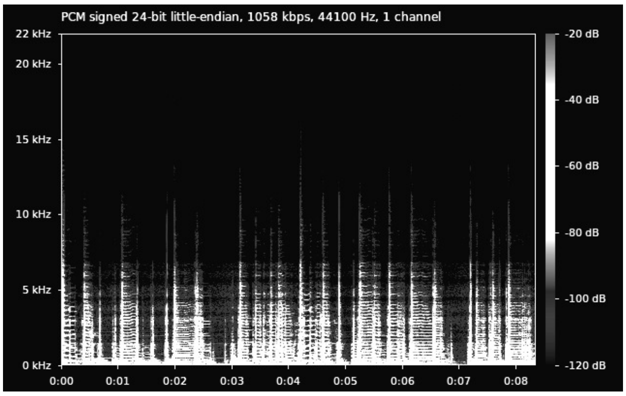


Figure 1A: Original sound

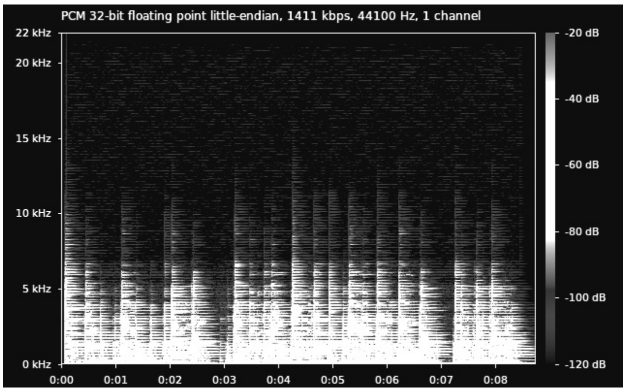


Figure 1B: Comb filter

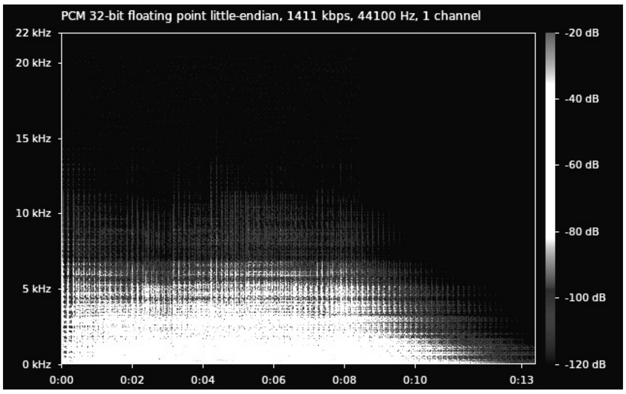


Figure 1C: Large echo

4. Effects with variable delay lines: *chorus*, *flanger* and *phaser*

As mentioned earlier, variable delay lines allow the change of the delay time while the audio signal is processed by the delay unity. This allows the creation of other effects based in delay lines: *chorus*, *flanger* and *phaser*. When we say that a delay line is variable, we are describing a unity of signal processing that has some element that varies constantly. In this case, what varies is the duration of the delay time. It can oscillate between a maximum and a minimum value. A low frequency oscillator (LFO) can implement this kind of effect. The LFO is used to control the delay time. Figure 2 shows an implementation of this model in Pure Data.

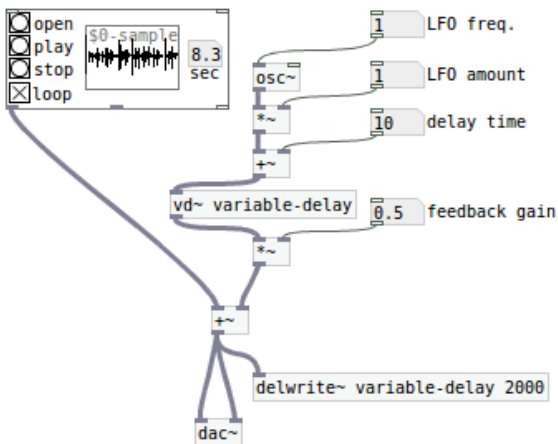


Figure 2: Model of variable delay with feedback

Based in this model of variable delay we can implement some kinds of effects. These effects have in common that same variable delay unity. However, each of them has some special features. One of these features is related to the variation of the delay time. In the case of *chorus*, for instance, this variation has to be set between 10 and 30 ms. In the case of *flanger*, the variation can occur between 1 and 20 ms, and in the case of *phaser* the LFO may vary from 1 to 10 ms.

In the sonograms in Figures 3A and 3B, we may visualize – mainly looking at the spectral content – the different patterns produced in the resulting sound signal by each of these different processes of time delay using variable delay lines. A clear distinction between these processes and the processes with fixed delay lines is that their main characteristic concerns changes in the temporal/morphological domain of the sound signal, except maybe in relation to the *comb filter* effect.

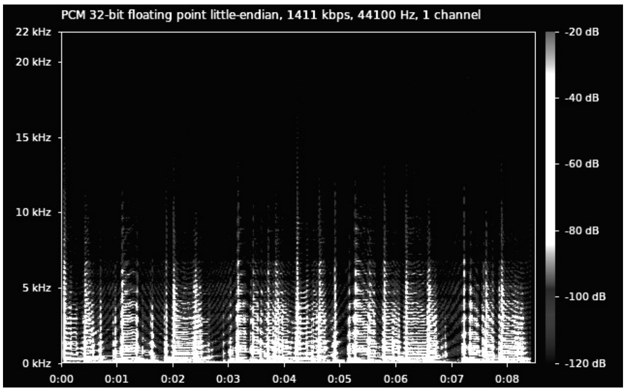


Figure 3A: Phaser

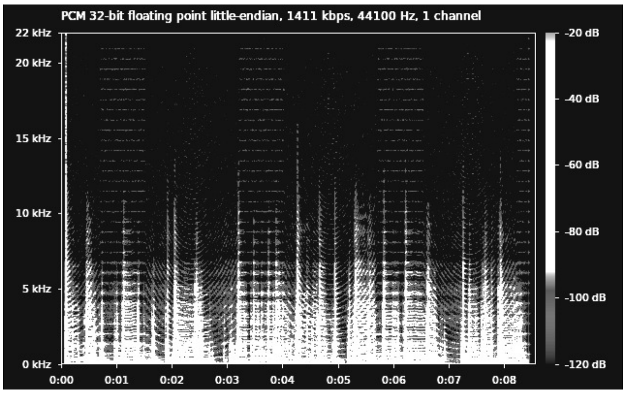


Figure 3B: Flanger

5. Conclusions

The understanding of the different types of signal processing, generated by fixed or variable delay lines, enhances the decision making process in situations when we face professional audio problems, from technical or aesthetical points of view. This can happen in a simple sound recording session, in an audio mixing station, or during the composition of an electroacoustic music that employs pre-processed or real-time sound effects.

It can be quite useful being able to differentiate between the results of processes such as *comb filter*, *phaser* and *chorus*. These effects result from delays that generate time displacement between their repetitions. This causes changes in the harmonic spectrum. The awareness of time elements can be useful in many situations, for instance in room reverberation, identification of obstacles in sound trajectory or recognizing phase cancelation in a recorded sound.

Therefore, in these cases, acknowledging the time prevalence in the situation can help the decision making to avoid attention only to the frequency domain, what can lead the sound engineer to use, for instance, a frequency filter to change the spectrum involuntarily. Indeed, we may create desirable effects with other strategies, like setting a short time displacement between similar tracks, as we use to do in voice or instrumental unison doubling. The result is an enlarged sound ambience and reinforcement of harmonic partials produced by constructive interference.

Similarly, we must be aware of changes in the spectrum domain generated by basic effects as reverbs and simple delays, as generated by delay lines, because they may be desirable or not. The superposition of repeated sound materials usually produces reinforcement of certain frequencies, generating an effect aesthetically desirable or just distortion. Our experience tells that this situation can happen when recording in a room with large reverberation. For the performer the sound seems nice but the signal for microphone caption can already be saturated at the source.

These experiments also demonstrate that the human perception of delays shows a double standard. Delays larger than 50 ms are interpreted by our brain as isolated repetitions while shorter delays just change certain frequency components of a single sound event.

What has been presented in this paper was a reasoning for the implementation of many types of signal processing with time delays. The kind of reasoning used was based in the creation of delay lines by digital means. This way we emphasize that there is not a single way to implement these processes, even digitally or using individual delay lines without feedback. The implementation with delay lines, in a digital environment, helped didactically the understanding of how are produced and how we perceive the effects based on time delays and that use less resources of digital processing.

References

Mauricio di Berardino. *Muselectron: Tutorial su PureData, Max/MSP e Axoloti.* Available at: https://muselectron.blogspot.com.br/p/pure-data.html Accessed in April of 2019.

Miller S. Puckette. *The Theory and Technique of Electronic Music.* World Scientific Publishing. 2006.

Jean-Michel Réveillac. Musical Sound Effects: Analog and Digital Sound Processing. Iste Editions, 2018.

Curtis Roads. *The Computer Music Tutorial.* The MIT Press, 1996.

3A: mAchine learning Algorithm Applied to emotions in melodies

Cláudio Gomes[1], Josue Barros[1], Marco Leal[1], Thiago Nascimento[1]

[1]Departamento de Ciências Exatas e Tecnologias - Universidade Federal do Amapá (UNIFAP)
Caixa Postal 261 – 68.906-970 – Macapá – AP – Brazil

josue.barros1986@gmail.com, {claudiorogerio, marcoleal, thiago.nascimento}@unifap.br

Abstract. *At every moment, innumerable emotions can indicate and provide questions about daily attitudes. These emotions can interfere or stimulate different goals. Whether in school, home or social life, the environment increases the itinerant part of the process of attitudes. The musician is also passive of these emotions and incorporates them into his compositions for various reasons. Thus, the musical composition has innumerable sources, for example, academic formation, experiences, influences and perceptions of the musical scene. In this way, this work develops the mAchine learning Algorithm Applied to emotions in melodies (3A). The 3A recognizes the musician's melodies in real time to generate accompaniment melody. As input, The 3A used MIDI data from a synthesizer to generate accompanying MIDI output or sound file by the programming language Chuck. Initially in this work, it is using the Gregorian modes for each intention of composition. In case, the musician changes the mode or tone, the 3A has an adaptation to continuing the musical sequence. Currently, The 3A uses artificial neural networks to predict and adapt melodies. It started from mathematical series for the formation of melodies that present interesting results for both mathematicians and musicians.*

1 Introduction

In all cultures music is present. Regardless of musical intentions, there was a process of musical creation. In this way, the musical composition is present in the part of a rich and particular process in the musical production of each environment. The composition can be the combination of sounds and silences within a space and time. This way, there are two important agents, the generator, or musician, and the listener [1].

The musician, during the written moment of composition, defines different ways the construction of his work, and style with a sonorous interaction. This construction is defined by their experiences and musical knowledge. Depending on the musical discourse, the musician can use various tools, instruments: guitar, saxophone, trumpet, flute; as well as digital instruments: synthesizers, percussion, and digital drums, etc.

With the emergence of the MIDI protocol, it allowed the development and appearance of countless digital synthesizers such as Casio, Korg, Roland, etc. There was also interest in musical research centers for the alternative development of modules or systems of sound synthesis or by MIDI protocol. Thus, with the development of instruments compatible with MIDI technology, the possibilities of output and the facilities with the control and verification in the construction and composition of melodies [2].

The technological advance allowed several solutions that help the musician in his compositions. The process of computer-assisted composition is a example in evolution that allows greater clarity and affinity of the musician to the computer [1, 3]. The LZ77 algorithm was used from parts of a musical base for the generation of melodies [4]. The LZ77 algorithm is a generic algorithm that uses the Euclidean distance concept to select the best fitness among the chunks or parts of each song in genotype generation. In addition, interesting melodies can be generated from inheritances from different parts of the music base. Thus, this model pre-processes to generate new melodies.

2 3A algorithm

The self-organizing map (SOM) is a learning algorithm used in some applications: image recognition, text, financial market, etc [5]. Basically, the SOM has input parameters and a single output. This learning consists of adapting numerous input possibilities, and without increasing the cost of processing, obtaining satisfactory results for each application. Each input parameter has specific importance to the evolution of algorithm learning, that it could vary temporarily. Thus, at each moment the inputs are evaluated to the algorithm learns and presents the best alternative.

The 3A utilizes a midi synthesizer device connected to a desktop or notebook computer by processing this data from the Chuck language to output audio or midi, as shown in figure 1. For this initial works, the 3A consists in a SOM with MIDI inputs and output. The inputs are 3 MIDI notes played temporarily, the output is MIDI note played as a melody to suggestions for accompaniment to musician in real time. Based on inputs, the 3A decides the harmonic field played by musician. The harmonic field is the map of learning and its neurons are the notes played temporarily, as seen in figure 2.

Figure 1: General view from 3A

In figure 2, the first note in SOM is the specific musical tone, in other words, in all lines the first note is the music field tonal and the other notes are part of musical tones. Each MIDI note played actives respective neurons, for example, if the G note was played, 3A active 5th neuron from the first line, 1st neuron from the second line, 4th neuron from the third line. After three sequential notes played, 3A identifies among the neurons which showed the highest synapse activated deciding a musical tone and generate MIDI notes. It is considered the highest synapse activated the horizontal line with more notes marked in SOM. In case of there is a duality of synapse identification, in other words, lines in same quantity of neuron activated, the algorithm selects the leftmost synapse neuron. At each moment, 3A decides the harmonic field and generates notes, that it is the melody for accompaniment.

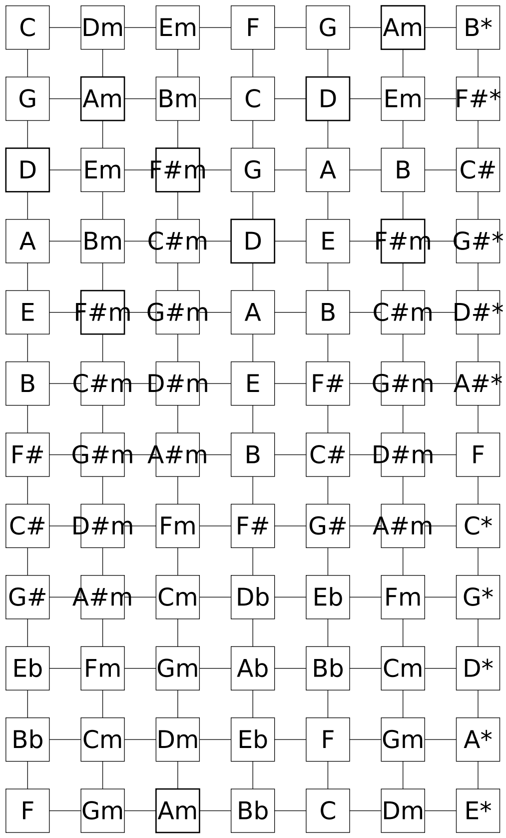


Figure 2: Self-organizing map of 3A

The 3A finds the musical tone among various to each line, and after that, suggest a note to accompany the input note in a same musical tone. This result can be transcribed in audio or MIDI. In figure 3 illustrates the machine learning model (3A) and still one of the activation neuron synapses.

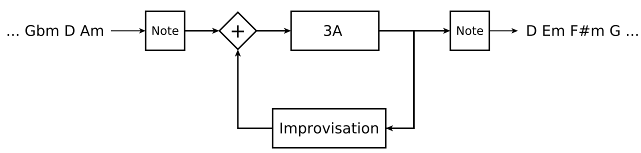


Figure 3: Structure of 3A algorithm

In first set of experiments, the sequence of notes or silences generated by 3A follows a normalized random model that vary in the temporal dimension between whole, half, quarter, and quaver. For tests, a score made by melody based on the series prime numbers was used, because this sequence does not have mathematically completely identified a pattern. The algorithm was implemented from the Chuck language [6]. The results obtained show that the sequence acquired by 3A varies according to the initiation of weights for each input parameter. Regarding the architecture of tests, it is being used models of embedded systems like the model Raspberry Pi[7].

References

[1] Jose D Fernández and Francisco Vico. Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, 48:513–582, 2013.

[2] International MIDI Association et al. Midi musical instrument digital interface specification 1.0. *Los Angeles*, 1983.

[3] John Strawn and Curtis Roads. *Foundations of computer music*. MIT Press, 1985.

[4] Manuel Alfonseca, Manuel Cebrián Ramos, and Alfonso Ortega. A fitness function for computer-generated music using genetic algorithms. *WSEAS Transactions on Information Science and Applications*, 2006.

[5] Teuvo Kohonen. *Self-organization and associative memory*, volume 8. Springer Science & Business Media, 2012.

[6] Ge Wang, Perry R Cook, and Spencer Salazar. Chuck: A strongly timed computer music language. *Computer Music Journal*, 39(4):10–29, 2015.

[7] Raspberry Pi Foundation. Raspberry pi. `http://www.raspberrypi.org`, may 2019. Accessed: 2019-05-05.

Sensitivity to Instrumentation of a Singing Voice Detector Based on Spectral Features

Shayenne Moura[1]*, **Marcelo Queiroz**[1]

[1]Computer Music Research Group – University of São Paulo
Av. Prof. Luciano Gualberto, 1171 - Butantã, São Paulo - SP, 05508-090

`shayenne@ime.usp.br, mqz@ime.usp.br`

Abstract. *Detecting voice in a mixture of sound sources remains a challenging task in MIR research. The musical content can be perceived in many different ways as instrumentation varies. We evaluate how instrumentation affects singing voice detection in pieces using a standard spectral feature (MFCC). We trained Random Forest models with song remixes for specific subsets of sound sources, and compare it to models trained with the original songs. We thus present a preliminary analysis of the classification accuracy results.*

1 Introduction

Singing Voice Detection, also referred to as Vocal Detection, is the task of identifying singing voice segments in a piece of audio containing a mixture of sound sources. This task is an intermediate step in many other tasks pertaining to Music Information Retrieval, among them singing voice separation[1] and melody transcription[2].

There are several approaches in the literature in order to identify segments with singing voice [3]. In this work, we use the feature engineering approach, i.e., we use features commonly related to voice processing tasks as input for machine learning algorithms to create a model classifier. This is a preliminary experiment which uses data augmentation based on training models with different remixes of pieces, and we present an evaluation of the classification accuracy using this approach.

In order to make the experiments automatic and easy to reproduce, the scripts for the extraction of audio descriptors are available on this github link[1], and the algorithm and evaluation are available on Jupyter Notebook files.

2 Methodology

Our goal is to compare the sensitivity of classifier models to instrumental remixes, by using a standard feature (MFCC) to perform singing voice detection. We used the MedleyDB dataset [4], which contains singing voice pieces with separate tracks for each instrumental source, and created alternative remixes by combining subsets of the original instrumental tracks.

We defined four families for these remixes, with progressively fewer instruments: (1) the original mix; (2) all monophonic instruments plus drums; (3) all monophonic instruments; and (4) only instruments playing the main melody. This creates an augmented dataset in which we want to measure the performance of singing voice detectors. It should be noted that all families include all vocal tracks, and the motivation is to verify if these new remixes would make the singing voice detection easier, as well as to obtain more data for training. A 5th family, consisting of purely instrumental remixes, i.e. original mixes without vocals, was also considered to provide more training data with non-singing voice examples, in an attempt to counterbalance the 71% rate of positive examples (i.e. segments containing singing voice) in the original data.

The ground-truth was based on instrument activations, as defined in the MedleyDB dataset [4]. We consider that a 960 ms segment has singing voice if at least 50% of its length (not necessarily contiguous) has singing voice. The types of singing voice included in our dataset are: male singer, female singer, male speaker, female speaker, male rapper, female rapper, and vocalists.

Audio features are calculated using 0.96 second segments, with 0.48 seconds overlap. Specifically, we use Librosa [5] 0.6.0 to obtain MFCCs of 40 coefficients using 10 ms segments, out of which we retain the first 13 coefficients (excluding the 0th coefficient); we then summarize every 96 segments (96*10 ms) using the following summary statistics: mean, standard deviation, median, delta and double delta, in order to preserve temporal context (feature dimensionality is 13 * n_statistics).

In the experiments we used Random Forest classifiers with 100 estimators, after considering as alternatives 10, 20, 50, 100 and 150 estimators, because 100 estimators consistently produced the best results in all experimental scenarios. To evaluate detection sensitivity, we conducted a first experiment to compare the classification accuracy of models trained and evaluated within each family of remixes. In a second experiment, we wanted to verify if trained models generalized well by progressively enlarging the training data: (A) training with only the original mixes (family 1); (B) training with original mixes plus alternative remixes that include vocals (families 1+2+3+4); and (C) training with original mixes and all alternative remixes (families 1+2+3+4+5 – including purely instrumental remixes).

*Supported by CNPq.

[1]`https://github.com/shayenne/VoiceDetection`

3 Evaluation

3.1 Dataset

The experiments were based on the MedleyDB [4] dataset. We selected all 61 tracks containing singing voice and split them into training and test subsets. The split was defined as follows: 80% for the training subset and 20% for the test subset, leaving 48 and 13 songs for the training and test subsets, respectively. This results in 21368, and 3874 960 ms audio segments for training and test, respectively.

To avoid the artist/album effect [6] in our classification experiments, we used the medleydb API[2] to ensure that the subsets do not share the same artists, i.e. if an artist falls into the training subset, all of her songs will be in the same subset.

3.2 Results

We used accuracy to evaluate the performance of the trained models. Figure 1 presents the results within each of the four remix families, i.e. training and testing the models within a single remix family. We can verify that singing voice detection becomes more accurate when training and evaluating with a reduced subset of sound sources, as compared to using all sources in the original pieces.

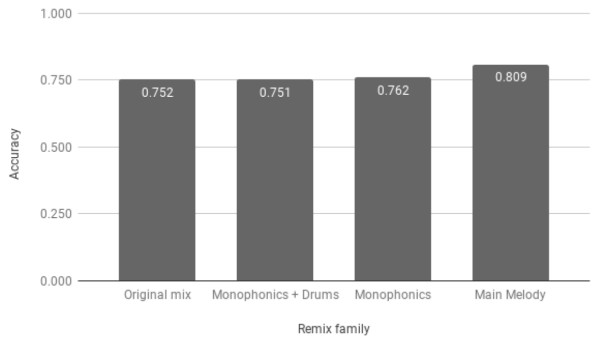


Figure 1: Accuracy using remix families

The results of generalization of our models are presented in Figure 2. In this experiment we trained the models using the three groups of training data discussed in the previous section, i.e. (A) only original mixes, (B) original + alternative remixes with vocals, and (C) original + all alternative remixes. Evaluation was made on the validation set of the original songs.

We see in figure 2 that accuracy decreased about 42% when using an augmented training set including vocal alternative remixes, in comparison to using only the original pieces in the training set. The results of classification accuracy training the model with all alternative remixes decreased yet a little bit more.

Our intuition to explain these negative results is that, even if the alternative remixes create specific contexts within which singing voice detection is slightly easier than in the original context (an interpretation endorsed by Figure 1), these contexts are possibly introducing too much

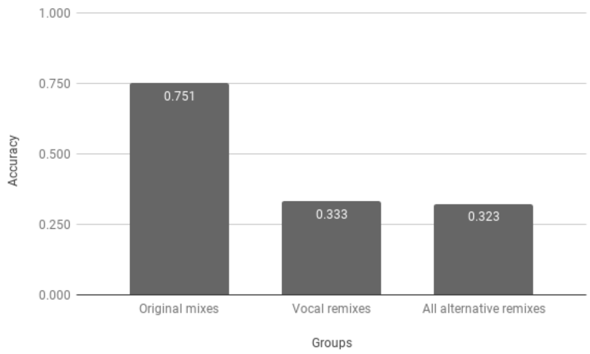


Figure 2: Accuracy in generalization of augmented training sets

dispersion in the generated MFCCs (which are well-known to reflect timbre aspects).

The confusion matrix for the last group, for instance, shows that the classifier has become substantially biased towards labelling segments as *not* containing singing voice (around 67% of all segments correspond to false negatives), suggesting that the MFCCs of the non-singing voice class overlapped most of the singing voice segments in this representation space. So, if data augmentation through diversifying instrumental variety is ever going to be useful in singing voice detection, other audio features, more directly related to the presence of voice, will necessarily have to be included.

Another observation derived from these results is the fact that adding pieces without vocals examples in the training set (in an attempt to balance the positive/negative examples) actually decreased the model ability to accurately classify singing/non-singing segments.

As future work in the direction of data augmentation techniques for singing voice detection, besides including more specific voice-related audio features in the representation space, we consider training the models with different mixes of songs, (e. g. woodwind sources, string sources), and using other classification models (as SVM, Neural Networks), within a specific set of instruments or music style/genre to be evaluated.

4 Conclusions

In this text we reported preliminary results of our experiments on evaluating the use of different instrumental remixes as a data augmentation technique for singing voice detection. We used Random Forest models to classify the singing voice segments from the MedleyDB dataset, using a standard audio feature (MFCC). Our results show that the remixes were not able to increase the classification accuracy in comparison to the use of the original pieces, but gave some insights for future improvement, such as evaluating the models trained with other groups of remixes based on instrumental families and combining MFCCs with other voice-related audio features.

[2] `https://github.com/marl/medleydb`

References

[1] Yipeng Li and DeLiang Wang. Separation of singing voice from music accompaniment for monaural recordings. Technical report, Ohio State University Columbus United States, 2005.

[2] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, Aug. 2012.

[3] Kyungyun Lee, Keunwoo Choi, and Juhan Nam. Revisiting singing voice detection: a quantitative review and the future outlook. In *19th Int. Soc. for Music Info. Retrieval Conf.*, Paris, France, 2018.

[4] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello. Medleydb: A multitrack dataset for annotation-intensive mir research. In *15th Int. Soc. for Music Info. Retrieval Conf.*, pages 155–160, Taipei, Taiwan, Oct. 2014.

[5] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto. librosa: Audio and music signal analysis in python. In *Proc. 14th python in science conference*, pages 18–25, 2015.

[6] B. Whitman, G. Flake, and S. Lawrence. Artist detection in music with minnowmatch. In *Proc. of the 2001 IEEE Signal Processing Society Workshop*, pages 559–568. IEEE, 2001.

Characterization of the sonority associated with woodwinds instruments through spectral analysis

Yubiry González, Ronaldo C. Prati

Center of Mathematics, Computer Science and Cognition - CMCC
Federal University of ABC – UFABC. Av. Dos Estados, 5001, Bangú
Santo André, 09210-580, SP, Brazil

Yubiry.gonzalez.17@gmail.com, rcprati@gmail.com

Extended Abstract

The sonority is one of the widely used definitions by musicians when trying to define the color or the balance of timbres associated with individual or groups of instruments, such as for ensembles or orchestras. Currently, those who use this term refer to subjective parameters associated with the "balance of tones," "amplitude of sound," and "mixtures of colors," using terms such as bright, opaque, robust, weak, among others, which they do not have a precise definition. Other theoretical musicians also associate sonority with relations between consonances, dissonances, and other harmonic aspects that may exist in musical compositions [1]. Other trends associated with electroacoustic music use the new synthesized sound resources to achieve mixtures of new colors, or sonority other than those obtained only with acoustic instruments.

The perception of sound is characterized by amplitude, tone, and timbre mainly, being related to physical-acoustic parameters, such as intensity, frequency, and the number of harmonics, respectively. Furthermore, it is well known that the number and relative intensity of the harmonics or secondary frequencies, corresponding to an individual sound, vary according to the type of instrument and its details in the manufacture (geometry and materials) [2]. In the same way, but the second-order, there is the corresponding to the own musical performance carried out by each interpreter which can originate, under the same acoustic conditions and the same instrument, different sonorities.

From the use of digital signal processing tools, these secondary or harmonic frequencies can be analyzed, allowing a direct gathering of information about the timbre or colors associated with each instrument. This process is useful for the characterization of the sonority due to its high relation with the timbre. Various studies have been carried out in this direction for woodwind instruments such as the transverse flute and the clarinet [3] that propose studying the timbre to extract information from the different possible colors in any musical instrument, proposing for its characterization a spectral centroid with the use of digital signal analysis [4].

For the present work samples of audio signals were obtained from three wooden wind instruments, such as transverse flute, oboe, and clarinet, which under optimal acoustic conditions executed melodic patterns of sounds with the same frequency, scales and musical fragments, following the methodology of González, Y. [5]. Then with the obtained audio samples, different spectral analyses were performed using Fast Fourier Transform (FFT), Spectral Power Density (DPE) and Spectrograms. Finally, ten professional musicians listened to the audio samples categorizing them subjectively in 6 terms related to sonority, which were brilliant, opaque, translucent, solid, soft, and metallic.

From the FFT and DPE, it was possible to quantify the number of harmonics present for a given fundamental frequency, proposing a coefficient defined as the weighted average of the

tones and intensities that expresses the magnitude of the harmonics present concerning the reference frequency or fundamental frequency. This indicator allowed to establish comparisons between sonorities or colors for each instrument, being the common factor in this family the associated with "soft" and "opaque" whose coefficients were 2.8 and 2.2 respectively. A peculiarity of these instruments is that, depending on the octave, the coefficient related to timbre changes by a factor of 3.5. This change gives greater versatility in sonorities for the instruments of this family.

From the spectrogram, the stages of sound evolution were analyzed. Each stage contributes to obtaining a defined sonority in a maximum time of 2 seconds for each sample, concluding that the attack and sustenance with which define the type of sound for each instrument to be listened to by professional musicians.

Finally, it is proposed to extend the method to other families of musical instruments for its generalization and potential application in the musical creation by sonorities as a compositional tool.

Acknowledgements: “This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES)

References

[1] Strasburger, H., and Parncutt, H. Applying psychoacoustics in composition: Harmonic progressions of non-harmonic sonorities. *Perspectives of new music*, *32*(1), pages 88-129. 1994.

[2] Aramaki, M., Besson, M., Kronland-Martinet, R., & Ystad, S. Timbre perception of sounds from impacted materials: behavioral, electrophysiological and acoustic approaches. In *International Symposium on Computer Music Modeling and Retrieval*, pages 1-17, Springer, Berlin, Heidelberg, 2008.

[3] Navalón Salas, I. et al. *Spectral analysis of the sound obtained with oboe tudels made with different materials. Editorial Spanish Acoustic Society, TECNIACÚSTICA 2015*, 46º Spanish Acoustics Congress. 9th Iberian Acoustic Meeting. European Symposium on Virtual Acoustics and Ambisonics, Valencia, 2015.

[4] Schubert, E., Wolfe, J., and Tarnopolsky, A. Spectral centroid and timbre in complex, multiple instrumental textures. In *Proceedings of the international conference on music perception and cognition, North Western University, Illinois*, pages 112-116. 2004.

[5] González, Y. Study and characterization of the timbre quality associated with wooden wind instruments. 48th Spanish Acoustics Congress; Iberian Acoustic Encounter; European Symposium on Underwater Acoustics Applications; European Symposium on Sustainable Building Acoustics: A Coruña 3-6, pages 1138-1146. Spanish Acoustic Society.2017.

Batebit Controller: Popularizing Digital Musical Instruments' Development Process

Filipe Calegario[1 2], João Tragtenberg[1 2], Giordano Cabral[1], Geber Ramalho[1]

[1]MusTIC / Centro de Informática / Universidade Federal de Pernambuco
Av. Jornalista Aníbal Fernandes, s/n - Cidade Universitária (Campus Recife) - 50.740-560 - Recife - PE

[2]CIIMUS / Instituto SENAI de Inovação para TICs / SENAI-FIEPE
Rua Frei Cassimiro, 88 - Santo Amaro - 50.100-260 - Recife - PE

fcac@cin.ufpe.br, tragtenberg@gmail.com, grec@cin.ufpe.br, glr@cin.ufpe.br

Abstract. *In this paper, we present an ongoing research project related to popularizing the mindset of building new digital musical instruments. We developed a physical kit and software intended to provide beginner users with the first grasp on the development process of a digital musical instrument. We expect that, by using the kit and the software, the users could experiment in a short period the various steps in developing a DMI such as physical structure, electronics, programming, mapping, and sound design. Our approach to popularizing the DMI development process is twofold: reducing the cognitive load for beginners by encapsulating technical details and lowering the costs of the kit by using simple components and open-source software. In the end, we expect that by increasing the interest of beginners in the building process of digital musical instruments, we could make the community of new interfaces for musical expression stronger.*

1 Introduction

This paper focuses on the technical details of the development process of a digital musical instrument. We believe that overcoming technical barriers by presenting them in a natural approach could open up possibilities for beginners. In sum, we hypothesize that when the technical barrier is presented as a natural step, the beginners can open their heads for new ideas.

The development of new digital musical instruments (DMIs) is an interdisciplinary process, in which each step relates to a specific mindset and requires a particular set of skills [1]. Mechanical structure, electronics, programming, mapping, and sound design can be loosely considered the development areas of a DMI.

Due to these different mindsets and skills, developing a DMI can become a laborious process. In this sense, the designer must manage different partnerships or become a polymath to achieve the musical artifact at the end.

Following the example of Arduino [2], which popularized the physical computing by simplifying the access of artists and designers into electronic prototyping, we believe that more straightforward and faster ways of developing new DMIs can contribute to more beginners experimenting ideas and, therefore, engagement in the DMI community.

By understanding the process, the users can adapt the technology to their needs, intentions, and contexts of use. The more people are experimenting and testing, the higher the chances of achieving great instruments ideas.

In this paper, we present a physical kit and software to serve as an entry point to DMI development. Our approach is to reduce cognitive load by encapsulating technical details. Besides, we consider that it is essential to make the kit accessible to broadening the audience. Therefore, we propose to use accessible and straightforward components, materials and manufacturing techniques, and open-source software, in order to reduce the costs and reach more people.

2 Related Projects

There is a considerable amount of hardware and software tools already available that could be used during DMI development process. Some examples are: microcontrollers environments (Arduino, Raspberry Pi, Beaglebone, Teensy), sensor kits (Infusion Systems, littlebits, makey makey), MIDI controllers (keyboard, wind controllers, percussion controllers), general-purpose programming languages (C, C++, Java), audio-oriented programming languages (CSound, SuperCollider, Chuck, Pure Data, Max/MSP), creative programming environments (Processing, openframeworks, Cinder, Scratch), applications for mappings (libmapper, iCon, OSCulator, juxion, Wekinator), and digital audio workstations (Logic Pro, Ableton Live, Pro Tools, GarageBand, Reaktor, Tassman).

Beyond those tools, there is a growing number of DMI's development toolkits aimed to reduce the technical barriers for musicians and designers [3, 4, 5]. However, we believe that these toolkits still presents an in-depth approach concerning the aspects of DMI development. For instance, Bela [1], a platform for musical interactions based on the BeagleBone board, focus on providing real-time processes and more natural ways of programming musical interactions. There is a significant number of examples that can help the users begin to develop their instruments, but they have to build their physical interface, mechanical structure, and electronics. There is no doubt that, with Bela, the users could have a profound, expressive result. However, for the popularization of DMI development, we find that it is more interesting to have an in-breadth approach.

[1]http://bela.io

3 Our approach

Concerning the beginners, we believe that an in-breadth approach contributes to a better understanding of the DMI development process. In other words, didactically, we consider that allowing the users to experiment a small, superficial amount of each stage of DMI technical development in a short period can make them better comprehend the process and identify what aspect excites them.

We intend to embed a little portion of the DMI development process into an object so the users can have an enactive experience when using it. We believe that the feeling of making an instrument with their own hands from zero to a functional prototype in a short period with little pitfalls is a productive way of teaching the complex and arduous process of developing a DMI.

The idea is to guide the users throughout the mechanical structure with MDF plates, screws and bolts; the electronics with a Arduino Leonardo, a shield, two knobs, one LDR (light sensor), six buttons, and six LEDs; the programming with a example code in Arduino IDE; the mapping and sound design with a patch in Pure Data. Figure 1 shows the disassembled kit.

Figure 1: Disassembled Kit

When assembled (Figure 2) and connected to the computer via USB cable, the kit becomes a MIDI controller that operates the parameters of a Pure Data patch. The buttons trigger samples and light of the LEDs. The two knobs change the time interval and the feedback amount of a delay effect. The LDR is related to the pitch parameter of an oscillator, which is gated by a dedicated button.

Figure 2: Assembled Kit

4 Workshops

The kit and software were informally tested in three workshops. Each one had twenty participants that worked in pairs. The kits were presented disassembled, and the participants had to follow a set of instructions in a PDF illustrated with photos. The pairs were encouraged to work without waiting for guidance from the workshop facilitator, that should be called only in the occurrence of an obvious error.

In two hours, the participants had to assemble, program, and play the Batebit Controller. After that, the participants answered a questionnaire with multiple choices and open text questions. One participant mentioned that he was expecting to see, but it was not covered the possibility to have a final product, instead of a prototype: *"How the project ceases to be a prototype and becomes a final product that I can take it to the stage?"*(P09). Moreover, two participants mentioned that the most critical aspect of the workshop for them was to have an overview of the entire process: *"The [most relevant part was] overview of what is happening in the entire process"*(P08) and *"The step-by-step of the process"* (P06).

5 Conclusions

The Batebit Controller is a simple in-breadth approach to embed the DMI technical development process into a kit and software that can be didactically experienced in a short period by beginners. The objectives are to be the entry point for popularizing the mindset of instrument making, reach a broader audience, and make DMI community stronger. The feedback of the workshop participants showed that the kit and software have the potential to inspired non-technical users to begin experimenting with DMI development. In future steps of the research, we plan to better understand the engagement of the participants on each phase of the technical development process.

References

[1] Ricardo Dal Farra, Carolina Medeiros, Filipe Calegario, Marcelo M. Wanderley, Jaime Olivier La Rosa, Jônatas Manzolli, Juan Reyes, Fernando Iazzetta, and Hamilton Mestizo. Research-creation in Latin America. In Simon Emmerson, editor, *The Routledge Research Companion to Electronic Music: Reaching out with Technology*. Routledge, 2018.

[2] Massimo Banzi. *Getting Started with Arduino*. " O'Reilly Media, Inc.", 2009.

[3] Andrew P Mcpherson and Fabio Morreale. Technology and Community in Toolkits for Musical Interface Design. In *CHI 2017*, pages 9–12, Denver, CO, USA, 2017.

[4] Jack Armitage and Andrew Mcpherson. Crafting Digital Musical Instruments: An Exploratory Workshop Study. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Blacksburg, Virginia, USA, 2018.

[5] Filipe Calegario, Marcelo M. Wanderley, Stephane Huot, Giordano Cabral, and Geber Ramalho. A Method and Toolkit for Digital Musical Instruments: Generating Ideas and Prototypes. *IEEE MultiMedia*, 24(1):63–71, jan 2017.

Art

"TecnoFagia: A Multimodal Rite"

Luzilei Aliel[1] Rafael Fajiolli[1] Ricardo Thomasi[1]

[1] Escola de Comunicações e Arte – (Universidade de São Paulo, São Paulo, SP, Brasil) Av. Prof. Lúcio Martins Rodrigues, 443 - Vila Universitária, São Paulo - SP, 05508-020 – São Paulo – SP – Brasil

{luzaliel}@usp.br,{rafael.fajiolli.oliveira}@usp.br,{ricardothomasi}@hotmail.com

Abstract:

This is a concert proposal of Brazilian digital art, which brings in its creative core the historical and cultural aspects of certain locations in Brazil. The term *Tecnofagia* derives from an allusion to the concept of anthropophagic movement (artistic movement started in the twentieth century founded and theorized by the poet Oswald de Andrade and the painter Tarsila do Amaral). The anthropophagic movement was a metaphor for a goal of cultural swallowing where foreign culture would not be denied but should not be imitated. In his notes, Oswald de Andrade proposes the "cultural devouring of imported techniques to re-elaborate them autonomously, turning them into an export product." The *Tecnofagia* project is a collaborative creative and collective performance group that seeks to broaden aspects of live electronic music, video art, improvisation and performance, taking them into a multimodal narrative context with essentially Brazilian sound elements such as:accents and phonemes; instrumental tones; soundscapes; historical, political and cultural contexts. In this sense, *Tecnofagia* tries to go beyond techniques and technologies of interactive performance, as it provokes glances for a Brazilian art-technological miscegenation. That is, it seeks emergent characteristics of the encounters between media, art, spaces, culture, temporalities, objects, people and technologies, at the moment of performance.

Program

Acts: A) Leituras do Interior n.1 / B) Morro da Antena (Urupema) / C) Choveu Tanto Que Virou Represa

Graph Composer: music composition from graph design

Pedro Arthur dos Santos Souza[1*], Vitor Guerra Rolla[1†], José Eduardo de Almeida Ayres[1‡], José Ezequiel Soto Sánchez[1§]

[1]Visgraf – Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110 – 22460-320 Rio de Janeiro, RJ

pasouza@impa.br, vitorgr@impa.br, jeaayres@impa.br, cheque@impa.br

Abstract. *The Graph Composer is an app that allows the user to compose music through the design of a graph. You can create or modify an existing graph, listen and modify the composition in real time. Insert new nodes and connect them, change the corresponding note by clicking over the node and selecting a new one from the scale, define it's duration over time and select a decoration to change the sound sequence.*

1 Background and motivation

A graph [1, 2] is a mathematical object consisting of a set vertices and edges that connect them. Each vertex can have several attributes or values depending on what is being modeled. The edges can have direction, in which case we talk about a directed graph.

In the Graph Composer, each vertex represents a note and it's duration over time. The edges connecting the vertices define a path over which the graph will be traversed, playing the notes on the sequence of connected vertices. The sound produced by each vertex has different forms according its decoration: a single note, chord or arpeggios.

For example, taking the following score, from Figure 1, it can be represented as a directed graph (Figure 2). Paths are the possible sequences of vertices that can be traversed in a graph. In this example, the graph has a unique possible path, but when there is more than one edge leaving a vertex, multiple paths are possible.

Figure 1: An example score

The app proposes to the public the possibility to compose music through a priori mathematical modeling, so they can both understand how a graph is represented and built as well as how the model is applied to musical compositions.

The app encourages the user to try and discover new graph constructs and forms that may sound pleasing or even unexpected.

*Supported by CNPq.
†Supported by FAPERJ.
‡Supported by FAPERJ.
§Supported by CNPq.

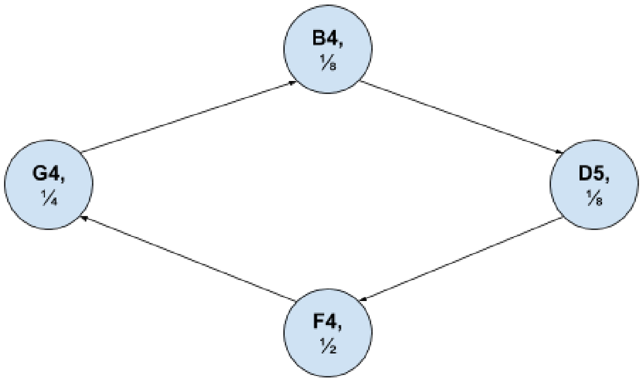


Figure 2: Directed graph representing the score from Figure 1

2 Model of the Graph Composer

The model used for the Graph Composer is actually applied in Computational Musicology research, an area where mathematics and computer science are applied to music. In research, the model is used to represent and analyze musical compositions. In the research, compositions are modeled as graphs to compare and extract different mathematical features. For example: the number of times that a transitions from one note to another is repeated, or the number of times that three (or more) different notes appear in the same sequence, among others.

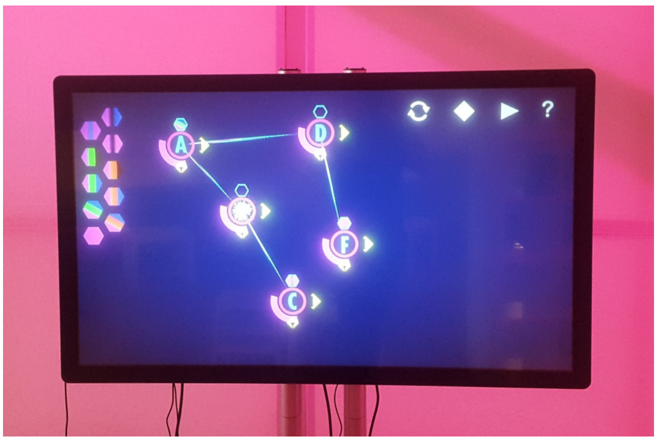


Figure 3: Setup of the app in a touch screen monitor

The app offers the possibility to compose music through the mathematical model of paths over graphs. The program walks paths along the graph from the root vertex through the directed edges until it finds a vertex with no outwards edges, when it returns to the root vertex. How it looks can be seen in the setup of Figure 3, and an example of the app in use can be seen Figure 4.

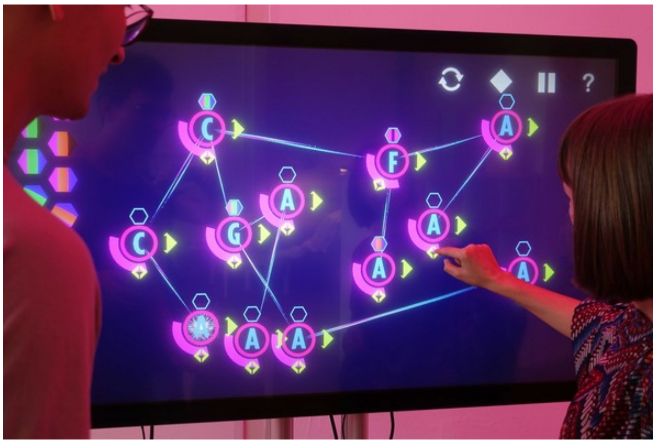


Figure 4: Usage of the app

In an empty setting, the user will find only the root vertex (one with no incoming edges), which he can grow adding new vertices and edges. The attribute of the vertex are: a note, a decoration and a time duration. The user can modify the note by clicking the vertex, the time duration using the slider in the side of the vertex and the decoration by dragging the available ones from the palette located in the left of the screen. The user can delete vertices dragging then out of the screen, and also delete edges by swiping their finger over it.

In the case that there exists more than one possible path in the graph, the program will choose one randomly. This means that when the program reaches a vertex with more than one outward edge it will choose one randomly. This creates what we call random walks through the graph. Since the music produced by each path could have different durations, the user may perceive this as some kind of arrhythmia

The app uses a minimalist design, with highly visual choices and options, but the user can still produce complex outcomes.

The app should lead the user to:

- Grasp the mathematical abstraction of a graph and its independence of its drawing. This is achieved by showing how the music played depends merely on the attributes of the vertices and the edges drawn, since the position of the vertices in the screen has no effect on the resulting music.
- Think the music as sequences of both simple and compound sounds. This is achieved by showing how several sounds can be represented by a unique symbol (chords and arpeggios) or that, conversely, simple musical representations can be enriched with this kind of musical structures. This resources are widely used by musicians to enrich interpretations of known music compositions as well as in improvisation.

3 Infrastructure and target audience

The infrastructure required for this application is very short: a touch screen monitor, a computer to run the app and speakers or a headphone. The setup should not take much space and it is intended to be located in a place where people can pass by and play with it for a few minutes.

Graph Composer was developed to be simple and encourage the user to try music in a game like way. The app is intended to reach a good range of people, besides being a very funny platform to learn some basic concepts of music and math, it can be seen in use on Figure 4.

References

[1] Reinhard Diestel. *Graph theory*. 5 edition, 2016.

[2] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*. Citeseer, 1 edition, 1976.

Per(sino)ficação

Fábio dos Passos Carvalho, Flávio Luiz Schiavoni , João Teixeira Araújo

[1] Federal University of São João del-Rei
São João del-Rei, MG

fabiopassosarquitetura@gmail.com, fls@ufsj.edu.br, teixeira.araujo@gmail.com

Abstract. *The bell's culture is a secular tradition strongly linked to the religious and social activities of the old Brazilian's villages. In São João del-Rei, where the singular bell tradition composes the soundscape of the city, the bell's ringing created from different rhythmic and timbral patterns, establish a language capable of transmitting varied types of messages to the local population. In this way, the social function of these ringing, added to real or legendary facts related to the bell's culture, were able to produce affections and to constitute a strong relation with the identity of the community. The link of this community with the bells, therefore transcends the man-object relationship, tending to an interpersonal relationship practically. Thus, to emphasize this connection in an artistic way, it is proposed the installation called: PER (SINO) FICAÇÂO. This consists of an environment where users would have their physical attributes collected through the use of computer vision. From the interlocking of these data with timbral attributes of the bells, visitors would be able to sound like these, through mapped bodily attributes capable of performing syntheses based on original samples of the bells. Thus the inverse sense of the personification of the bell is realized, producing the human "bellification".*

1 Introduction

The bell culture is a secular tradition that was and still is strongly linked to the religious and social activities of the old villages of the Brazilian colonial period. The ringing bells, created from different rhythmic and timbral patterns, are semiotic objects by establishing a language capable of transmitting various types of messages to the local population[1].

In São João del-Rei, Minas Gerais, the focus of this work, the unique bell's tradition present in the composition of the soundscape of the city and the strong affective and identity relation of part of the population with these sonorous instruments, conferred to this the nickname of "city of the bells" or "city where the bells speak"[2]. This affection for bells produces, on a certain scale, a personification of these, making them something more than inanimate objects[3].

Different periods in the catholic calendar have different rhythms and every church has a different bell set up, that comes from a single bell to up to three of them that can be combined by the composers to create all the different patterns present in the city. Based on this tradition and as a manner to emphasize this connection in a contemporary artistic way, it is proposed the installation called: PER (SINO) FICAÇÂO.

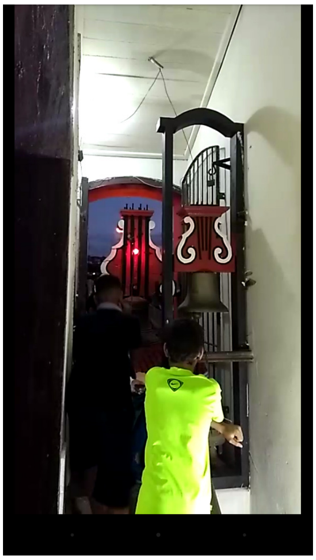


Figure 1: The vision inside the steeple.

2 The installation

In this piece, the audience is invited to get inside the steeple and be a bell of the church. When coming into the space, some projections of different steeples of the city's churches are projected and the bells are quite. A calm soundscape of the city is the only sound heard in the place. When getting inside the steeple, the bells starts ringing in different patterns and rhythms and the city soundscape is completed by the sound of the bells.

The nature of the composition varies depending o the visitor's features. A camera system using Computer Vision algorithms captures the visitors body and the weight, size, and shape of the visitor's body and these attributes are mapped to a new bell with these personal features. Every visitor inside the chamber can change drastically the sound of the church putting a different bell in the musical pattern and creating a different music. The combination of visitors can create different rhythms and patterns and up to 5 bells can ring together in our steeple.

The sound of the bells are synthesized using the visitors physical attributes and mixed with a background soundscape[4] that can bring the city around the church to the installation creating a more realistic atmosphere based on the day time.

Completing the installation, images from the steeple and along the bell towers are projected and also controlled by the visitors data to create an immersive environment and an unique experience. These images and videos, captured in real steeples of the city's churches, fol-

Figure 2: A Steeple with two bells.

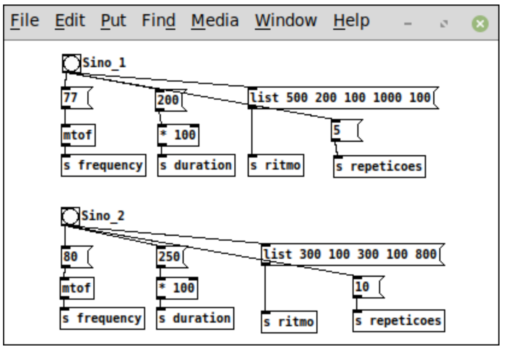


Figure 4: PureData Bells Prototype

low the rhythm defined by the visitor's particulars and can also be influenced by the movement of bodies inside the chamber. In addition to the proposed visual projections, in some moments the absence of visual stimulus inside the cabin will also be explored, with the aim of providing a sound-only experience, where the public is encouraged to concentrate only in the sound it produces and the performance interaction with the other users of the installation. This experience is based on the Accusmatic principles, where only sound is the central musical aspect and, in this case, producer of interaction.

3 Technical Aspects

Regarding the technical aspects used in the installation, this work has two main parts (Figure 3). Initially, information such as audience height and width is collected from a Python based on the OpenCv library to process the image provided from the webcam. Thus, a message of the type -height,width- is sent through the Open Sound Control (OSC) protocol to a Pure Data patch.

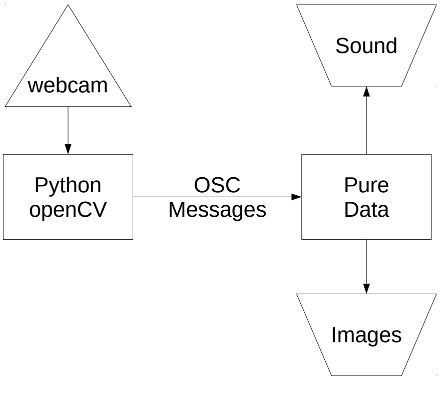


Figure 3: The information flow in our system.

Upon receiving OSC messages, the application implemented in Pure Data is responsible for issuing bell samples (Figure 4). This Pure Data programming merges different layers of sound including samples, soundscapes, the wind in the steeple and the synthesized bells, along with some images and videos .

4 Acknowledgments

Authors would like to thanks to all ALICE members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq and FAPEMIG.

References

[1] André Guilherme Dornelles DANGELO and Vanessa Borges BRASILEIRO. Sentinelas sonoras de são joão del rei. *Belo Horizonte: Estúdio*, 43.

[2] Thiago Morandi. Sentinela sonora. `https://www.youtube.com/watch?v=9xPTkbXPAfo`.

[3] Jason Barroso Santa Rosa, Rodolfo Magalhães, and Alessandra Oliveira. Entoados. `https://www.youtube.com/watch?v=sBswNbF3HTc&t=279s`.

[4] R Murray Schafer. *The soundscape: Our sonic environment and the tuning of the world.* Simon and Schuster, 1993.

O Chaos das 5

Flávio Luiz Schiavoni, Adilson Siqueira, Rogério Tavares Constante, Igino de Oliveira Silva Junior, Thiago de Andrade Morandi, Fábio dos Passos Carvalho, João Teixeira Araújo, André Lucas Nascimento Gomes, Frederico Ribeiro Resende, Luísa Lugon-Moulin Brazil Santos, Avner Maximiliano de Paulo, Carlos Eduardo Oliveira de Souza, Allyson da Cruz Rodrigues, Igor Alves Estevam Andrade, Aneliza Rodrigues Prado, Gabriel Lopes Rocha , Patrick Veniali da Silva, João Lucas Ruas Teixeira, Aretha Brito de Lima, Bruna Guimarães Lima e Silva, Ívyla Nascimento Silva, Wanderson Andre Martins Pereira, Rafael Gomes de Oliveira, Octávio Ferreira Deluchi Cerqueira da Silva, Valdir de Siqueira Santos Junior, Matteo Vinicius Alves Recinella

[1] Arts Lab in Interfaces, Computers, and Everything Else - ALICE
Nucleus of Art and Sustainability - NAST
Federal University of São João del-Rei
São João del-Rei, MG

fls@ufsj.edu.br

Abstract. *"O Chaos das 5" is an audiovisual digital performance. The guideline of the performance is inspired by Alice, from Lewis Carroll book - Alice in the Wonderland, as a metaphor to take the audience to a synthetic and disruptive wonder world. The concept of the performance is to conceive the possibility to the audience to interact through digital interfaces creating an immersive and participatory experience by combining three important layers of information (music, projections and gestures) through their cellphones. Once that the audience members take part of the show on an immersive aspect, there is no stage or another mark to limit the space of the performers and the audience.*

1 Introduction

O Chaos das 5 is a piece of collective creativity developed by the Transdisciplinary research Group (GTRANS) from the Federal University of São João del-Rei involving the ALICE (Arts Lab in Interfaces, Computers and Else) / Orchidea (Orchestra of Ideas) group from the Computer Science Department and the ECOLAB / Movère group from the Scenic Arts Department.

Figure 1: Performers and the audience

The guideline of the performance took Alice, from Lewis Carroll book - Alice in the Wonderland [1], as a metaphor to take audience to a synthetic and disruptive wonder world. The audience members could take part of the show and there is no stage or another mark to limit the space of the performers and the audience.

Three layers of information were used to create an immersive experience to the audience combining music, visual and gesture by the means of technology. Five musicians located around the space create the base of the sound using DMIs and over processed electric guitars. The audience members could participate of the sound using their cellphones and accessing a set of web DMI developed to this performance.

The visual layer used three projectors and some software developed in our lab to create images in real time. Here, the aesthetic approach is to break the Black Box [2] and show to the audience what is the technology behind the scene. The code of the software sometimes is also projected, like in a live coding [3] performance, and two programmers were changing the code and the visual on the fly. Images took by the audience members with their cellphones are also used in this layer and webcams and image processing in real time completed the visual set up.

The gestural layer is performed by artists interacting physically with the audience members in the space. In the beginning of the show it was probably not easy to identify who were the artists and who were audience members. This performatic artists merged gestural score and improvisation to interact among them and with audience members and their participation became more clear during the show.

2 The plot

We started the performance with the projection of a countdown clock and an invitation to the audience take part of the performance accessing a website. In the website, the audience members could find some instructions and instruments that could be used during the show. They could also inform their names and upload a picture to register their participation.

The first part of the performance is a synthetic universe, the dive in the rabbit hole, among synthetic images and infinite glissandos that remembers Metastasis from Xenakis. During this part, the performers that were

Figure 2: The live coding room

among the audience members started acting in a reverse form, revelling their selves as part of the performance and taking attention to them.

The free falling finishes in a second part, a disruptive experience in the real world. A territory battle in the city where people try to exist and register this existence guided this part of the performance. We projected a noisy sequence of pictures of graffiti and other urban scenes while a city soundscape completed a saturated urban scene. Performers started painting their selves using stickers and brushes and the audience members were invited to do the same.

To escape the reality and the tension of the second part, a third part took the audience to a surreal experience, calming down until the end. The performers, tired of the second scene, start a slow dance in front of the distorted projection of themselves.

At the end, like a credit film, a projection presents the name of all members of the performance including the audience members that filled their name and picture in the website.

Figure 3: Performers interacting with projections

3 The concepts

The main line in this performance is about questioning realism and surrealism in the contemporary world when facing brute technology interacting with our bodies. The show update the notion of reality putting on the scene the precision of the digital media and the uncertain rhythm of machines in our lives. The concept of immersion in Virtual Reality implies in using machines hiding them from the people resulting in human beings being exposed to digital stimulus without noticing the presence of machines. In ubiquitous computing, the presence of machines is so deeply felt that machines can interact directly with humans beings being part of our reality.

That is the concept explored in our performance in our visual and musical layers. All the computers and live coders are present in the scene showing to the audience their screens and codes. Linux terminals commands, coding being compiled in real time, all these artifacts serve to break the barrier of the computer invisibility during the performance. Everything can be seen but it certainly does not make it easier to be understood. The unveiled machines increase discomfort but not necessarily increase the level of understanding. In the middle of all these unveiled technology, the performers and the audience deals with improvisation and the effect of technology in our daily lives.

Our musical layer is inspired in three contemporary composers: comes from glissandos inspired in Metastasis from Xenakis and Shepard tones, soundscapes composed to create a urban interaction environment, inspired from Murray Schafer, and a surrealist calm inspired by Olivier Messiaen.

Figure 4: Performers, live coders and audience

This improvisation environment lead us to several questions: A mistake is really a mistake? When is it considered part of the show? What is the limit to improvisation and a programmed manner to work? Our narrative is uncertain and a distance from what is hermetic and finished is what we try to reach as our main theoretical goal.

4 Acknowledgments

Authors would like to thanks to all ALICE and Movère members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq and FAPEMIG and the support of the PROEX - UFSJ.

References

[1] Lewis Carroll. *Alice Adventures in Wonderland & Through the Looking-Glass*. CreateSpace Independent Publishing Platform, 2018.

[2] Vilem Flusser. *O Mundo Codificado (Em Portuguese do Brasil)*. Cosac & Naify, 2007.

[3] Nick Collins, Alex McLean, Julian Rohrhuber, and Adrian Ward. Live coding in laptop performance. *Organised sound*, 8(3):321–330, 2003.

Body Building Music: The *Kinase* installation

MONNAZZI, João Paulo Silva [1] *, FARIA, Regis Rossi Alves [2]

[1] Escola de Comunicação e Artes da Universidade de São Paulo Av. Prof. Luciano Gualberto, 158, tv,3 – 05508-900 São Paulo, SP.

[2] Escola de Artes, Ciências e Humanidades, Universidade de São Paulo Rua Arlindo Bettio, 1000 – 03828-000, São Paulo, SP

joao.monnazzi@usp.br, regis@usp.br

Extend Abstract

Thinking on the congruencies between music and sports, we propose with this art installation some novel paths and connections for music production in a little explored field, in the interdisciplinarity with sports.

Some similarities in the acting of musicians and athletes, such as the need of technical domain through discipline and practice. A musician who wants to develop her/his technical skills needs to follow a hard routine of practical studies, focusing in improving motor abilities with the proposing to play the piece in the better way possible.

This process has a close proximity with the athlete's during their preparation. Hours of intense practice to improve some motor skills that can enable them to improve their performance. The disciplines can be interpolated in a way that we can argue: there is always something physical on a music interpretation, as well as there is always something artistic in a sport competition.

In the inner area between art/music and sports some modalities are easier to verify this symbiosis, as in the choreographic sports. These modalities are evaluated by both physical and artistic parameters. Our work focus in a particular sport modality that has a part of scoring which is evaluated through a choreographic routine: The Bodybuilding.

1. The sport and the Body Building Music engine

Bodybuilding is a sportive modality evaluated using artistic parameters [1]. The body of athletes are analyzed by symmetry, conditioning and volume of muscle mass. The competitor who has the most symmetric, lean fat free and body muscle, wins. During the bodybuilding competition, one of the most expected moment is the solo presentation, when the bodybuilder performs a choreographic routine with musical accompaniment. In this part of the championship, the evaluated parameters are creativity and synchronicity. Our interactive system supports this part of the show.

Body Building Music (BBM) is an interactive music system that allows the user to control a musical performance and its parameters through gestures derived from the bodybuilding sport. The system was enabled by musical controllers that use Kinect (camera sensor based) as a motion sensor to trigger or reshuffle sound materials [2].

The idea of the system is to provide the user with a double experience. The first is to compose music in real time in a way not requiring a prior knowledge of musical knowledge, and the second is to experience and get to know the bodybuilding sport through the poses required in the official championships.

The system was structured in functional levels and implemented with a mixed computer programming strategy, using Pure Data (Pd) and C-based routines that permit to integrate its physical components, comprised of a *gestural sensor* and a *computer* responsible for gestural processing and music generation. The functional rationale is attached to the gestural poses from the bodybuilding sport [3].

The mapping strategy used was a gesture for each sound (one-to-one mapping) [4]. Eight mandatory poses from bodybuilding championship were chosen to control inputs and output of musical layers and two continuous gestures were programmed to control a EQ filter and the dynamics of the musical sounds played back. We can then separate the control process in three levels, as below.

First level: the music structure interactive control level. In this level we can control the input or output of musical layers previously recorded. Each one of the 8 poses of bodybuilding is to deliver a different musical material. By posing, the user is enabled to mount a musical piece through sequencing of poses matched to musical parts. The order in which the poses are made will affect or to change the musical structure.

Second level: the effect (fx) control level. This level can be actived any time during the process by hanging the right hand open for more than two seconds. After this movement the user is enabled then to control the central frequency of a semi-parametric filter affecting the sound equalization. Through a continuous movement of his/her mirrored hands, the user can close or open the spectral coverage of the filter.

Third level: the dynamic level control. This control level is activated by hanging the left hand open for more than two seconds. After accessing this control stage, the user can modify the dynamics of all active playing musical parts. By raising both hands, the dynamic level increases, as well as moving both hands down will make the volume to decrease.

These three control levels were implemented to enable the user to experience a real time composition and music modulation process through a non-musical classical gesture, and he/she can still learn about bodybuilding and try some of its poses, experiencing the particularities of that sport.

The hardware system is composed by a camera sensor (Kinect), a laptop, a DAW (digital audio workstation) and a sound monitoring system (see fig. 1). It captures the performer gestures that will be mapped accordingly to a table of poses.

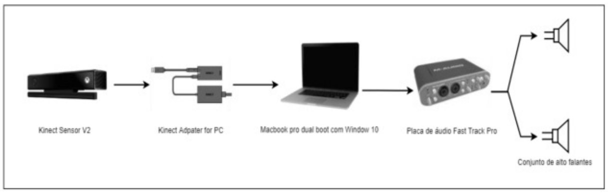


Figure 1: Hardware connections diagram

The user's movements are captured by the Kinect, compared with a repository of poses previously built. If the program recognizes a valid movement it releases an OSC message to a Pure Data (Pd) patch, where the musical generative actions are programmed, synthesizing the audio and delivering it using a stereo sound system. Figure 2 below shows a flowchart of the processes integrated in the system.

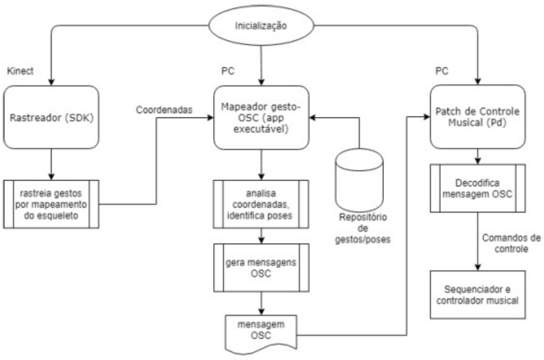


***Figure 2: BBM system flowchart ***

2. The Kinase musical creation

For the system to function as intended, bridging with musical real-time creation, it was composed an interactive musical piece [5], that would permit to use all the BBM system function. This was how Kinase was born, as a musical interactive composition-based musical creation.

Kinase is a modular piece, layer-based, which is instantiated dynamically through a real-time generation of audio controlled by gestures. During the system usage, the user can add just one of the eight available musical layers at a time, and it thus will be overlapped with other layers obeying a given synchronicity, so as to permit that musical parts with different execution times are triggered in the beginning of the right measures.

Figure 3 shows the piece structure and the instrumentation. The input order of each layer defines the music structure as well as the moment of activation of each layer. Other possible actions include the activation and control of a filter that will act on all activated tracks, serving as an artistic resource for performance. In the same way, the user can activate and control the dynamics of the piece through continuous gestures.

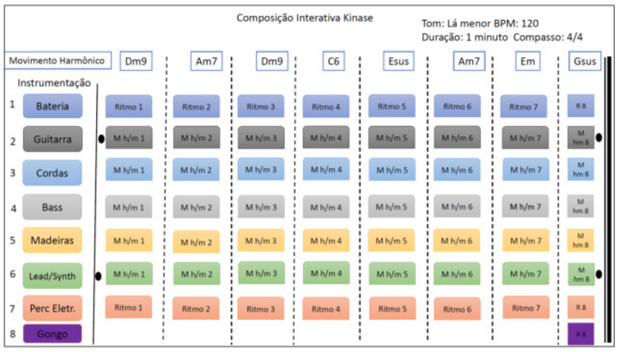


Figure 3: The Kinase structure

3. Final remarks

The installation is proposed to be run during the conference, in a public place, and be available for people interested in trying up with the system. A table describing the poses that control the creative process will be available, as well as some guidance to train users on how to use it.

References

[1] A. Locks and N. Richardosn. *Critical Reading in Bodybuilding*. Routledge, Nova York, 2012.

[2] MONNAZZI. J. and FARIA. R. *O uso do Kinect como instrumento digital em sistemas interativos para performance*. *15°* Annals Congressional of Audio and Engineering Society of Brazil AES.

[3] MONNAZZI. J. and FARIA, R. *Body Building Music: Sistema interativo para criação musical controlado por performance coreográfica esportiva*. Master's dissertation. São Paulo, 2018.

[4] J. Rova, M. Wanderley, S. Dubnov and P. Depalle. *Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance*. In: Proceedings of the AIMI International Workshop, Genova, Associazione di Informatica Musicale Italiana, pages 68 – 73, 1997.

[5] M. Sweet. *Writing Interactive Music for Video Games*. Pearson Education. Crawfordsville, 2015.

Black Lives Matter

Avner Maximiliano de Paulo[1], Carlos Eduardo Oliveira de Souza[1], Bruna Guimarães Lima e Silva[2], Flávio Luiz Schiavoni[1], Adilson Siqueira[2]

[1] Arts Lab in Interfaces, Computers, and Everything Else - ALICE
[2]Nucleus of Art and Sustainability - NAST
Federal University of São João del-Rei – UFSJ São João del-Rei - MG - Brazil

avnerpaulo.mg@gmail.com, carlosbolin@hotmail.com

limaesilvaguimaraes@gmail.com, fls@ufsj.edu.br, negrados@ufsj.edu.br

Abstract. *The Brazilian police killed 16 people per day in 2017 and 3/4 of the victims were black people. Recently, a Brazilian called Evaldo Rosa dos Santos, father, worker, musician, and black, was killed in Rio de Janeiro with 80 rifle bullets shot by the police. Everyday, the statistics and the news show that the police uses more force when dealing with black people and it seems obvious that, in Brazil, the state bullet uses to find a black skin to rest. Unfortunately, the brutal force and violence by the state and the police to black people is not a problem only in this country. It is a global reality that led to the creation of an international movement called Black Lives Matter (BLM), a movement against all types of racism towards the black people specially by the police and the state. The BLM movement also aims to connect black people of the entire world against the violence and for justice. In our work, we try to establish a link between the reality of black people in Brazil with the culture of black people around the world, connecting people and artists to perform a tribute to the black lives harved by the state force. For this, the piece uses web content, news, pictures, YouTube's videos, and more, to create a collage of visual and musical environment merged with expressive movements of a dance, combining technology and gestures. Black culture beyond violence because we believe that black lives matter.*

1 Introduction

Black Lives Matter (BLM) is a movement that began in 2013 founded by three black activists (Alicia Garza, director of the National Domestic Workers Alliance, Patrisse Cullors, director of the Coalition to End Sheriff Violence in Los Angeles and immigrant rights activist Opal Tometi) as a form of demonstration against the acquittal of neighborhood watchman George Zimmerman in the murder of 17-year-old student Trayvon Martin. However, in 2014, after the murder of two more black youths, Michael Brown and Eric Garner, the movement gained strength and became nationally recognized. Today, the BLM movement has become an organization that aims to fight not only against police brutality, but also against the conditions that oppress blacks around the world, economically, socially and politically [1] [2]. The movement can be found in the Internet with the hash tag #BlackLivesMatter.

With the current neofascist wave and the growth of right-wing governments around the world, we are seeing an increasingly stimulus to the return of movements such as the Ku Klux Klan, which bring the black population of the world into concern for possible setbacks in their rights. In Brazil, it is not different. Brazil is the non African country with the biggest afro descendant population in the world and one of the last country in the world to abolish slavery. Nowadays, a black person is 3 times more propense to be killed and most part of the murders in the country happened to afro Brazilians. Marielle Franco, a black city councillor from Rio, the only black female representative and one of seven women on the 51-seat council was killed in 2018. The killers were two former policeman. According to Human Rights Watch, the police force in the state of Rio de Janeiro, Brazil, killed more than 8,000 people between 2005 and 2015, 3/4 of them were black men. At the same time, the African culture strongly influenced the Brazilian culture and most part of the traditional Brazilian music and rhythms can be considered black music.

Certainly, the black people influenced not only the Brazilian culture but several traditional music around the world, mostly associated with drums, chorales and voices and almost never associated with technology. That is our tribute to the BLM movement, a performance that can be a utterly new form to put black people in the music, associating black skins with technology in a contemporary music scenario.

2 The Performance

The Black Lives Matter performance is a tribute to the BLM movement. According to the BLM's site, the purpose of the movement is to connect the black people around the world to seek justice and fight against the violence and racism, mainly by the police and the state. We wanted to approach this discussion to the computer music area. Our performance is a connection between music, dance and projections in a kind of improvisation dance and composition where computers and technology is putted together. Then, acting like an protest act.

Two musicians play in the computer a sum of YouTube videos, news and other online content, creating a collage of different historical moments of black people in art and news. This disruptive and sometimes awkward piece allows to put together anonymous and famous people, Samba and Blues, CNN and Globo. Merging layers of YouTube videos, the musicians create an visual and sonic environment playing in the same stage B.B. King and Naná Vasconcelos, with famous speeches and reporters.

Figure 1: Initial experiments of the performance.

The musical layer of the spectacle is mainly inspired by Pierre Schaeffer's *Musique concretè* [3], were we work with clippings from various videos, making a collage based on a pre-established score. The result is a set of samples being played in parallel, generating something chaotic and fun, due to the diverse origins of the clippings. The visual part intents to be a "white box", showing through two projectors, everything that's on the computer screens, with the tool and score open, side by side, and the hole process of the collage, being generated live.

Behind this projections, the performer realizes a dance, inspired in Anne Teresa de Keersmaeker & Michele Anne de Mey's Fase (1982) and Steve Reich's Music [4], addressing some points of the aesthetics worked on this performance, where the main idea is for the actress to dance with her shadow (see Figure 3), based on slightly off-synchronized music, in order to bring about changes in performance and music in a gradual way, to make the exchanges of context for the public almost imperceptible.

Figure 2: The projections over the performer's skin, creating a shadow on the wall.

3 Program Notes

The "Black Lives Matter" movement is about violence inflicted on Black communities by the state and vigilantes. Through the usage of the web as a tool to make collage art with online content, we try to approach this social question to the reality of Brazil, without disregarding its origin, merging this two realities and putting on the scene the hard reality of black people and also several artists and important figures for these cultures. Beyond that, which is shown by the projections and the sound, we have the presence of a corporal performer, completing the show with the shadow produced by her dance and the light of the projections reflecting on her skin.

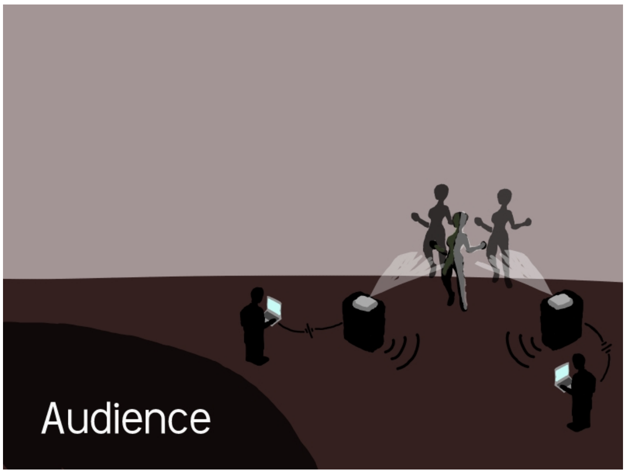


Figure 3: Stage plot

4 Technical Details

- Performance time: 23 minutes.
- Instrumentation: Computers and projectors.
- Musicians
  - Avner Maximiliano de Paulo
  - Carlos Eduardo Oliveira de Souza
- Performer
  - Bruna Guimarães Lima e Silva
- Intended venue: Cultural Center of UFSJ
- Setup time: about 30-40 minutes.
- Technical requirements
  - 2 Speakers
  - 2 Projectors

5 Acknowledgments

Authors would like to thanks to all ALICE and NAST members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq and FAPEMIG.

References

[1] Russell Rickford. Black lives matter: Toward a modern practice of mass struggle. In *New Labor Forum*, volume 25, pages 34–42. SAGE Publications Sage CA: Los Angeles, CA, 2016.

[2] Alicia Garza, O Tometi, and P Cullors. A herstory of the# blacklivesmatter movement. *Are all the women still white*, pages 23–28, 2014.

[3] Carlos Palombini. Machine songs v: Pierre schaeffer: From research into noises to experimental music. *Computer Music Journal*, 17(3):14–19, 1993.

[4] Renate Bräuninger. Structure as process: Anne teresa de keersmaeker's fase (1982) and steve reich's music. *Dance Chronicle*, 37(1):47–62, 2014.

Iterative Meditations: The use of Audio Feature Extraction Tools on Acousmatic Music Composition

Aluizio B. O. Neto[1]*

[1]CEGeME Lab – Universidade Federal de Minas Gerais

aluizioneto@ufmg.br

Abstract. *This piece explores some possibilities of using Music Information Retrieval and Signal Processing techniques to extract acoustic features from recorded material and use this data to inform the decision making process that is intrinsic to music composition. By trying to identify or create sound descriptors that correlate to the composer's subjective sensations of listening it was possible to compare and manipulate samples on the basis of this information, bridging the gap between the imagined acoustic targets and the actions required to achieve it. "Iterative Meditations" was created through an iterative process of listening, analyzing, acting and refining the analysis techniques used, having as end product the musical piece itself as well as gathering a collection of tools for writing music.*

1 Program notes

A collage of different soundscapes from urban and natural environments and musical material, assembled by means of digital sound processing techniques. Iterative Meditations is the result of an experiment with the gathering of audio samples and the use of sound feature extraction tools on the creative process of writing an acousmatic piece. The composer intended to develop a continuous repetitive cycle of listening, analyzing, acting and refining the analysis techniques used, having as end product the musical piece itself as well as a set of tools for music writing.

2 Extended Abstract

2.1 Background

Acousmatic music arose from the novel possibilities brought by recording technologies. Pierre Schaeffer's train [1] carried a sonic revolution that made 20th century's concept of "Music" a very hard thing to define. Using sound as raw material and manipulating it directly on a physical level by means of analog and/or digital devices allowed for a new way of interacting with this art form and new modes of listening, separating sound from its source. That doesn't mean the sources were not identifiable, but that they were physically detached from the performed piece, being reproduced through loudspeakers by means of amplified electrical currents.

Audio feature extraction plays a major role on almost all fields related to Music Information Retrieval. The piece presented here was written as a initial effort in researching the use of audio feature extraction techniques in the creative process of acousmatic music composition.

*Supported by CAPES.

Samples were previously recorded or gathered from different sources, being it mono, conventional stereo or binaural. The material was digitally manipulated with the use of free and open source audio software tools and the whole process was informed by data extracted from the original acoustic material. The use of sound descriptors allowed to select, transform and organize sounds based on feature similarity or an intended acoustic goal. Low level descriptors such as spectral flux, spectral centroid, MFCC's, estimated f0 and others were combined in order to find parameters correlated to the composer's perception, and used to guide the decision-making process of creating the piece. In this first experiment, higher level descriptors were intentionally avoided in order to keep the understanding of the process closer to the physical level, since the intention was to find simple transformations that could achieve the targeted outcome imagined by the composer. Visualization software was used to combine the extracted features, waveform and spectrogram into a cohesive graphical way of analyzing the piece while writing it. Figure 1 shows a screenshot of a section of it with two channel spectrogram combined with spectral centroid and amplitude follower curves.

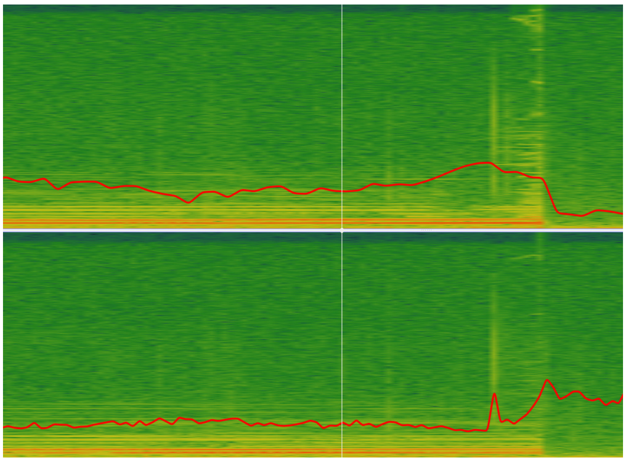


Figure 1: Screenshot of feature visualization

2.2 Methodology

Seems appropriate to describe the methods used in this project by starting from the beginning. The seed for the writing of this piece was a recording of a sharp tone extracted from a glass object hitting a metallic surface. The first sound heard in the final mix was created by combining and manipulating different samples that were chosen by matching their sound features to those of the original tone. First, the sensation of pitch from the sharp glass sound was correlated to an approximate fundamental frequency of the sound (maximum common divider from the most prominent pure tones present in the original sample). From this f0-related data and the spectral distribution of the original sound, some notes from the piano were chosen in order to keep approximately that same sensation of pitch and brightness, while enhancing the timbre of the original sound and bringing it to a more familiar musical sound. A similar process happened when working with the short-term, transient characteristics of that first sound. The attack-time and spectral decay of the original sample were used in order to select and manipulate other sounds that would be combined with it afterwards to achieve the imagined acoustic goal. These newly selected samples as well as their transformed counterparts were than used as raw material for the subsequent part of the piece, and so forth, until its completion, in what could be described as a "fractal-like" or recursive procedure.

Audio analysis and visualization tools were used to achieve the methods described in the previous paragraph. The author mainly used VAMP plugins [2] installed in the Sonic Visualiser package[3]. Secondary tools developed in Matlab® and Python programming languages were also used for data extraction and audio manipulation throughout the writing of the piece. A subset of such secondary tools comprise some methods found as part of the Expan Toolbox [4], developed at the CEGeME Lab at UFMG, and an implementation of the "Deterministic plus Stochastic Model" proposed by Xavier Serra [5].

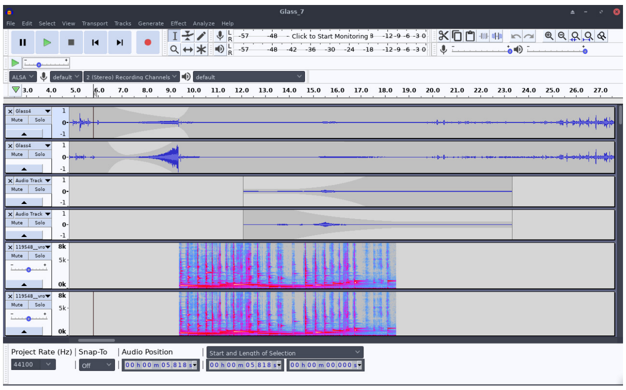


Figure 2: Multi-track visualization and editing

Audacity® [6] was chosen as a host for basic editing and the use of plugins, as well as a platform for simple multi-track waveform and spectrogram visualization (Figure 2). Some tools for spectral blurring, time stretching, frequency shifting and envelope modeling were used through its interface in order to speed up the process and automate some of the work. To achieve complete control of the processes applied, only free and open source plugins were used.

2.3 Conclusion and Future Work

This is a proof of concept piece, in which the author tried novel computational approaches for informing the creative process and combining different methods and tools in the creation of a musical piece. The techniques developed and systems used here are to be described in greater detail and published as a complete article in the near future. The development of more pieces intended to further explore the ideas presented here is also planned for the forthcoming months. Ideally, these tools and methods should be combined into a cohesive package that would bring together all the stages of the process in a unified interface that could ease the dialog between the steps of the creation cycle described in the first paragraph of this text (Program Notes).

References

[1] Pierre Schaeffer. Étude aux chemins de fer. *Cinq études de bruits, Studio d'Essai, présenté le*, 5, 1948.

[2] Chirs Cannam. The vamp audio analysis plugin api: A programmer's guide. *Availble online: http://vamp-plugins.org/guide. pdf*, 2009.

[3] Chris Cannam, Christian Landone, and Mark Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1467–1468. ACM, 2010.

[4] Thiago Campolina, Maurício Loureiro, and Davi Mota. Expan: a tool for musical expressiveness analysis. In *Proceedings of the 2nd International Conference of Students of Systematic Musicology*, pages 24–27. IPEM-Institute for Psychoacoustics and Electronic Music Ghent, 2009.

[5] Xavier Serra et al. Musical sound modeling with sinusoids plus noise. *Musical signal processing*, pages 91–122, 1997.

[6] D Mazzoni and R Dannenberg. Audacity [software]. *The Audacity Team, Pittsburg, PA, USA*, 2000.

Workshops

Workshop proposal: introduction to automatic audio classification

Tiago Fernandes Tavares[1]*

[1]School of Electric and Computer Engineering – University of Campinas
Av. Albert Einstein, 400 – 13083-852 – Campinas, SP

tavares@dca.fee.unicamp.br

Abstract. *This hands-on workshop comprises essential techniques for digital signal processing and machine learning. Participants will use the Python libraries librosa and scikit-learn as support to build an automatic audio classification system. The workshop will use explorations in toy problems to approach theoretical aspects. Later, it will discuss practical issues for building a scientific applications in the field.*

1. Program

1. Introduction to Python3 and visualization of audio signals in the time domain,
2. Frequency-domain visualizations and spectrograms,
3. Calculating and visualizing spectral shape descriptors,
4. Mapping audio signals to vectors,
5. Machine learning algorithms and pipelines (normalization, KNN, SVM),
6. Evaluating audio classification systems,
7. Discussion on the problem of audio tagging.

2. Target audience

- Up to 20 participants.
- Previous experience with Python is desirable, but knowledge on any computer language is quickly transferable to Python.
- Participants with no experience in programming are welcome, but their progress will likely be a little slower.

3. Required Infra-Structure

This is a hands-on workshop based on computer programming. As such, it requires one computer per participant. The computers can run on any operating system, and must have Internet access. The workshop will use online-hosted resources, hence no local configuration is required.

Ideally, a dedicated computer laboratory (with desktop computers) will be used. Optionally, it is possible to ask participants to bring their own laptops, as long as a reliable wi-fi connection is provided.

*Supported by CNPq.

Procedural Music in Games

José Eduardo de Almeida Ayres[1]*, Pedro Arthur dos Santos Souza[1]†, Vitor Guerra Rolla[1]‡, Luiz Velho[1]

[1]Visgraf – Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110 – 22460-320 Rio de Janeiro, RJ

jeaayres@impa.br, pasouza@impa.br, vitorgr@impa.br, lvelho@impa.br

Abstract. *This workshop will bring to the audience an introduction to the Chuck audio programming language, to the Unity game engine within a hands-on experience how one can use such technologies to achieve a new level of immersion through procedural generated sounds responding to game events and visual effects. The workshop is intended to a broad audience ranging from programmers to ones with little to no knowledge in the field.*

1 Background and motivation

With the advent of video games in the eighties, sounds (and music) were a crucial piece of immersion giving depth to the game-play. Even if nowadays graphics are on the spotlight, sounds are still a significant area in development. New technologies, for instance, Chunity [1], allows developers to provide fantastic sound experiences through game-play.

Audio can deeply connect to our emotions contributing to the immersion of the player in the game. Video games became a broad entertainment platform that gave birth to new technologies which facilitate the developer to implement his/her ideas. One of these technologies is the Chuck programming language which offers total control over sound synthesis and playback to the developer.

The Chuck programming language provides a wide range of tools for generating sounds algorithmically and in conjunction with the Unity game engine [2], which provides a simple environment for game development. One can achieve great results without requiring much knowledge of such technology. Therefore these are perfect for a hands-on workshop where anyone interested in the subject can learn and produce good results.

Our research group is continuously experimenting and developing new ideas through these technologies (Chuck, Unity), having already presented various workshops on the subject and developed a few apps. As said by [3], even if you use commercial software to create sounds, ChucK has new ways to control and interact with them. It's like being able to put a new engine in your car.

2 Program

The workshop program will be essentially practical. We start by showing you how to use the basic features of Unity: installation, interface, game objects, assets, and scripts in C# (1 hour). Then we will show the basic features of ChucK programming: the standart library, the math library, oscillators, sampling, and envelopes (1 hour). Chuck is not difficult because it looks a lot like C. In the next 2 hours, we will develop a game project and create the background music and sound effects with Chuck and Unity.

*Supported by FAPERJ.
†Supported by CNPq.
‡Supported by FAPERJ.

3 Infrastructure and target audience

The infrastructure required for the workshop is, in a way, very simple. The required physical space can be a classroom or an auditorium with a projector and speakers, we're going to use our own computers to show examples. We will need internet connection, the software configuration will be part of the workshop program. If the participant wants, he can follow the examples in his computer, but we will do everything on-the-fly, and everyone can follow without a computer.

The workshop was developed to be simple and to reach a good range of people. To take more advantage of the workshop, it is good that the participant has some previous knowledge in programming, but it is not mandatory. The target audience of the workshop is people of all ages who have an interest in the development of computer music and its applications, i.e. computational sound synthesis, creation of synthetic musical instruments, on-the-fly audio programming, and production of sound and music in games. In this vast audience are contained many professionals, like musicians, programmers, designers, and mathematicians. Anyone who is interested in using the tools described as means of producing audiovisual experiences. Substantially, everyone at SBCM is welcome in our workshop.

This workshop will contribute to SBCM by presenting, promoting, and training its participants in the discipline of audio and video game programming. Concerning technology, this event will add to the dissemination of Chuck and Unity programming interfaces.

References

[1] Jack Atherton and Ge Wang. Chunity: Integrated audiovisual programming in unity. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 102–107, 2018.

[2] Daniel V. de Macedo and Maria Andréia Formico Rodrigues. Experiences with rapid mobile game development using unity engine. *Comput. Entertain.*, 9(3):14:1–14:12, November 2011.

[3] Ajay Kapur, Perry R Cook, Spencer Salazar, and Ge Wang. *Programming for musicians and digital artists: Creating music with ChucK.* Manning Publications Co., 2015.

Keytalks

The First Brazilian Symposium on Computer Music presents Brazilian computer music potentials - Caxambu, MG, 1994

Mauricio Alaves Loureiro

[1]CEGeME - Center for Research on Musical Gesture and Expression – UFMG
Av. Antônio Carlos, 6627 – 31270-010, Belo Horizonte, MG

mauricioloureiro,@ufmg.br

Abstract. *The main objective of this talk is to report on the First Brazilian Symposium on Computer Music, which occurred in August 1994, at the city of Caxambu, Minas Gerais, promoted by the UFMG. The meeting occurred one year after the creation of NUCOM, a group of young academics dedicated to this emerging research field in Brazil gathered as a discussion list. This quite exciting and fancy event at Hotel Gloria in Caxambu was able to imposingly launch the group to the national, as well as to the international academic community. First, due to the excellency of the event's output and its daring program, that included 34 selected papers by researchers from various institutions from Argentina, Brazil, Canada, Denmark, France, Hong Kong, Mexico, UK, and USA, five lectures an two panels of discussion offered by researchers from the most advanced computer music research centers all over the world. The program also included eight concerts, two of them featuring traditional music, such as Bach, Mozart, and Brazilian music.Six computer music concerts presented 48 selected compositions submitted to the symposium. Second, as the symposium happened as apart of the 14th Congress of Brazilian Computer Science Society (SBC), the excellency of its output was able to attract the interest of SBC's board of directors. They invited NUCOM to integrate the society as a Special Committee, which are sub-groups of SBC dedicated to specific computer science topics. At the end of the description, this report aims at raising questions, arguments, and debates about today's format of NUCOM meetings, considering more seriously the interdisciplinary character of the methodologic approaches adopted by the field. Interdisciplinarity should be pursued by striving to contaminate a growing number of different topics of musical sciences, as well as of other research fields.*

1 The history

In 1990 Wilson de Pádua Paula Filho, a professor at the Computer Science Department of the Federal University of Minas Gerais (UFMG), created the Laboratory for Analysis and Synthesis of Image and Sound (OASIS). In 1991, I was appointed lecturer at the Music School of UFMG and joined OASIS in the following year. Chairing the music section of the Twentieth Fourth Winter Festival of UFMG in 1992, I invited Arcela to conduct a computer music workshop, Music and Artificial Intelligence, at Wilson's laboratory OASIS. He accepted the invitation, provided the whole team of his Lab, Márcio Brandão, Anselmo Guerra de Almeida, and Geber Ramalho could come along.

In July 1993, Geber Ramalho set up a computer music discussion list on the Esquina das Listas server of the State University of Campinas, launching the Brazilian computer music group, the NUCOM. The list included Arcela's team, Eduardo Miranda, in a doctoral residency at Edinburgh University, Wilson de Pádua, and me, among others. One of the topics discussed on the list was the need for organizing a computer music meeting in Brazil in order to map what would be going on in this field within the Brazilian academic community.

In August 1993, I participated for the first time on the congress of ANPPOM, the National Association for Research and Graduate Studies on Music, held at the Federal University of Rio de Janeiro, where I reported the ongoings towards the emergence of computer music as an institutionalized academic research field. During the meeting, I also had the opportunity to get to know the musicians and researchers involved in computer music by conducting studies and producing artwork. Convinced about the urgency to organize a Brazilian computer music meeting, I began to consider the possibility of a short event at the UFMG in the second semester of next year (1994). The event would happen at the School of Music of the UFMG. Although very well located in the middle of the city of Belo Horizonte, the School of Music could not offer adequate equipment for the meeting. Wilson pledged to make sure his department provides all the necessary equipment for the event. The first step would be to submit a funding project to research agencies, CNPq, the National Research Council, and FAPEMIG, the Minas Gerais State Research Foundation, for which I counted on Wilson's vast experience on this kind of fundraising.

Coincidently, Wilson's department was organizing the 14th Congress of the Brazilian Computer Science Association – SBC, to take place at the balneary city of Caxambu, Minas Gerais, around the same date we were planning our meeting. It was Wilson who had the brilliant idea to consider realizing our meeting as a sub-event of the SBC Congress. His colleague Nívio Ziviani, the congress' general chair, was moved by the idea of having a section dedicated to music, which, according to him, would bring a "certain charm" to the congress. Nívio introduced me to the organizing committee, to which I presented the project, after shortly describing the history and the state of the art of computer music, as well as the emerging isolated efforts in Brazil. The committee welcomed the proposal unanimously. Some of them were very attracted by the idea of having art production side by side to science. The benefit to our meeting was evident. Other than the immediate visibility the congress would offer to NUCOM's initial efforts, the congress organization made available to us all the necessary infrastructure, equipment and logistic for housing, transport, registration and above all, and the possibility of launching the first Brazilian Computer Music Symposium at the super fancy Caxambu Hotel Gloria within the advantageous contract facilitated by the over 1200 participants of SBC congress. At the end of the meeting, Nivio, as a great connoisseur of classical music, jokingly set a condition for the congress to shelter us: the realization of a classical music concert for the opening ceremony. I took it seriously and committed to making it happen, as long as the congress organization would provide house and transport for a group of musicians of the UFMG School of Music. The joking deal yielded not only a Congress opening concert where we played Mozart, Bach, and Brazilian folk music but also made it possible for our symposium to have a resident musical ensemble. Formed by high-level professional musicians, the group performed computer-mediated compositions, specially composed for the symposium, as described below. About that time, I received positive responses from the funding agencies for the projects I submitted. Upon such unexpected massive support from SBC and research agencies, we could afford to extend the event to a three-day full symposium and finance at least one international key-note speaker. The acceptance at the NUCOM list was overwhelming.

In June 1993, I organized the visit of Robert Willey, an American composer and computer music researcher of the Centre for Research in Computing and the Arts (CRCA), at the University of California San Diego (UCSD). Bob Willey stayed for a short residency at UFMG campus, lecturing at Wilson's lab and the School of Music. He also collaborated with the Grupo de Música Contemporânea da UFMG, an instrumental ensemble of faculty members dedicated to exploring new forms of expression involving improvising, acting, dancing, and computer-aided live performance, coordinated by me since its creation in 1992. With Bob Willey, I established contact with Chris Chafe from the Centre for Computer Research in Music and Acoustics (CCRMA) at Stanford University. I invited him for the key-note talk of the Symposium. He not only accepted the invitation immediately but also proposed to take along John Chowning plus three other researchers from CCRMA. Later in 1993, I was invited for a one-month-long residency sponsored by the Rockefeller Foundation, at the Centre for Research in Computing and the Arts (CRCA). During this residency, Chris Chafe organized a meeting with several researchers from both CCRMA and CRCA. The Argentinian electroacoustic composer, Francisco Kroepfl, director of the Laboratorio de Investigación y Produccion Musical (LIPM) at Buenos Aires, also participated in this meeting. The Foundation had sponsored the development of computer music massively at both research centers, other than offered financial support for international interchange promoted by them, including Kroefpl's lab, benefited with researchers' mobility, as well as equipment acquisition. Further financial support for international interchange with Latin America was also discussed in this meeting. Chris Chafe proposed to extend the Foundation's support to foster the field in Brazil, an attempt already made the year before with the Music Department of the Universidade de São Paulo (USP), through Prof. Marcos Branda Lacerda. Among other planned proposals, it was decided that the Foundation would fully finance the participation of CCRMA and CRCA members at our symposium, other than sponsoring short residencies for Brazilian faculty members dedicated to computer music, at CCRMA and CRCA. Among them, Sergio Freire and Gilberto Carvalho from the School of Music of UFMG and Aluizio Arcela from the Computer Science Department of Universidade de Brasilia (UnB) attended the program.

Rockefeller Foundation support guaranteed the impressive participation of some of the world's most prominent computer music researchers: David Jaffe, Fernando Lopez-Lezcano, Dexter Morril and Xavier Serra from CCRMA, Robert Willey from CRCA and Franciso Kroepfl from LIPM. Chris Chafe was not able to attend the symposium but came to the second edition in the following year. With the funds raised at Brazilian research agencies, we were able to invite Stephen Travis Pope, from the Centre for New Music and Audio Technology (CNMAT) at the University of California Berkeley.

2 The Symposium

Over one hundred people enrolled for the First Brazilian Symposium on Computer Music, happened between 3 and 5 August 1994, in conjunction with the Fourteenth Conference of the Brazilian Computer Society. Constituted the scientific committee Aluizio Arcela (Computer Science Department, University of Brasilia - UnB), Eduardo Reck Miranda (Edinburgh University), Geber Ramalho (Universite Paris VI), Jamary de Oliveira (School of Music, Federal University of Bahia - UFBA) and Wilson de Padua Filho (Computer Science Department, UFMG). Joined me at the concert committee, Conrado Silva (Music Department, UnB), Francisco Kroepfl (LIPM, Buenos Aires), Robert Willey (CRCA, University of California San Diego).

On the eve of the I SBCM opening ceremony, Ivan Moura Campos, the Brazilian federal secretary for Information Technology, came to us enthusiastically commenting on our proceedings. He called us for a meeting with Ricardo Reis, the President of SBC, to convince us that our Symposium met all requirements to be granted the status of an SBC's Special Committee. These committees are sub-groups of SBC dedicated to specific computer science topics, such as computer architecture, database, computer graphics and image processing, educational informatics, artificial intelligence, robotics, among others. Aluizio Arcela confirmed with much joy that we should certainly accept the invitation. I then invited Ricardo Reis to be joining the SBCM opening cerimony, where he made public SBC's interest in hosting our group as a Special Committee. In the morning of August 5th, NUCOM's members met SBC's President to formalize the invitation. Present at this meeting were Aluizio Arcela, Rodolfo Caesar, Régis Faria, Osman Gioia, Ricardo Jacobi, José Augusto Mannis, Jônatas Manzolli, Silvia Matthews, Eduardo Miranda, Axel Mulder, Jamary Oliveira, Wilson de Paula Filho, Conrado Silva, Francisco Kroepfl, Robert Willey, Bernadete Zagonel. We approved the creation of the Special Committee on Computer Music of the Brazilian Computer Society unanimously. In 1995, during the II SBCM, promoted by the Federal University of Rio Grande do Sul, in the city of Canela, the SBC board of directors approved the creation of the Computer Music Special Committee, for which I was appointed the coordinator.

Subsequent SBCMs maintained the participation of researchers from the most prominent computer music research center from all over the world. In 1995, the Rockefeller Foundation financed the participation of Chris Chafe (CCRMA) and Richard Moore (CRCA), as promised in 1994. The II SBCM was also able to invite Heinrich Taube (ZKM, Karlsruhe, Alemanha) and Marc Leman (Institute for Psychoacoustics and Electronic Music - IPEM, Belgium). Subsequent SBCMs brought Barry Vercoe (Media Lab, MIT), the Bell Laboratory computer music legend Max Mathews, the electroacoustic composer Denis Smalley (City University of London), Curtis Roads (CREATE, University of California Santa Barbara), Jean-Claude Risset (Université Marseille) and others.

2.1 Paper Presentations

Thirty-four selected papers were presented by researchers from various institutions from Argentina, Brazil, Canada, Denmark, France, Hong Kong, Mexico, UK, and the USA grouped into six subjects: (1) Systems and Languages for Sound Synthesis, Signal Processing, and Sound Transformation; (2) Music Notation Systems; (3) Systems and Languages for Composition; (3) Musical Analysis and Education; (4) Artificial Intelligence, Psychoacoustics, and Cognitive Models; (5) Performance, User Interface, and Instrument Design.

2.2 Lectures

The symposium's special guests presented five lectures:

- *The Well-Tempered Object: Musical Applications of Object-Oriented Software Technology*, presented by Stephen Travis Pope (CNMAT – University of California Berkeley).
  Chosen to be a key-note talk to the XIV Congress of Brazilian Computer Science Society, Pope spoke to over 1200 participants. He showed how computer music makes use of computer science, favoring the creation of new technologies. As a practical example, Pope established a link between object-oriented software technology and the essential features in computer music technology.
- *Os Caminhos da Pesquisa em Computação & Música no Brasil*, presented by Aluizio Arcela (Universidade de Brasilia).
  Arcela described his research efforts from the 1970s to the present, presenting the development of his "Time Trees" method for composition, an approach that at least partially owes its originality to Arcela's isolation from international research centers.
- *Computer Music at LIPM - Foregoing Music Productions in Argentina*, presented by Francisco Kröpf1, Miguel Calzón, and Carlos Cerana (LIPM, Buenos Aires).
  They described the development of a pioneering South American computer music laboratory, presenting equipment developed at LIPM, which has consistently proven ingenious, despite the recent use of relatively high-tech devices such as NeXT computers.
- *Composing for Interactive Instruments-Conductor, Soloist, and Improvisation Paradigms*
  David Jaffe (CCRMA – Stanford University).
  David presented three possible ways to interpret and perform with interactive instruments, using the MIDI Baton developed at the CCRMA. The third, in particular caught special attention for its ability to seek and find the balance between interpretative and creative gestures, a fertile area for further development of live-performance instruments.
- *Current Directions in Digital Synthesis: Physical Modeling and Spectral Modeling*
  Xavier Serra (CCRMA – Stanford University).
  Despite the technical complexity of some issues, Serra's exposition was clear. Using sound examples, he demonstrated the musical possibilities of using physical and spectral modeling as compositional tools.

2.3 Panel Discussions

- *Perspectives for Educational Programs in Computer Music in Brazil*
  Aluizio Arcela (UnB), Fernando Lezcano and Xavier Serra (CCRMA), Francisco Kropfl (LIPM), Jamary Oliveira (UFBA), Mauricio Loureiro (UFMG), Robert Willey (CRCA)
  Different proposals for interdisciplinary educational programs at the graduate level on Computer Music were described, with a discussion on the adoption of such programs in the Brazilian context. Undergraduate computer music curricula was also discussed. An international instructional program via the INTERNET, currently developed by CCRMA and CRCA, was also presented. An outline of the current situation of institutional financial support for educational and research projects within the Music area opened a discussion on funding perspectives for computer music projects.
- *Main Research Efforts in Computer Music*
  Conrado Silva (UnB), David Jaffe and Dexter Morrill (CCRMA), Eduardo Miranda (University of Edinburgh), Rodolfo Coelho de Souza (São Paulo), Stephen Travis Pope (CNMAT)
  Current lines of Computer Music research at the world's leading research centers were discussed, as well as possible directions in the next future. Some research projects carried out by Brazilians were described, pointing out their historical importance for the development of Computer Music in the country. Joint projects and

perspectives for financial support by international foundations, the current direction, and criteria for project evaluation were also discussed.

2.4 Concerts

The symposium put together eight concerts:

- The opening ceremony of the XIV SBC Congress began with a concert directed to all participants. The program included: the Suite No. 2 in B minor by J. S. Bach, with Mauricio Freire playing the flute and Oiliam Lana playing the harpsichord and conducting; the Quintet for Clarinet and String Quartet in A Major by Mozart, with Mauricio Loureiro at the clarinet.
- A second concert featured Brazilian music: Villa-Lobos, Mignoni, Valsas, Choros, Sambas and Baiãos.
- Six computer music concerts presented the 48 selected compositions from the submissions in two sessions each day. On every afternoon tape only music was played. Another concert, dedicated to live performance, was presented every evening, with the participation of the contemporary music ensemble of UFMG, the Grupo de Música Contemporânea (GMC): Benjamin Coelho (bassoon), Dilson Florencio (saxophone), Edson Queiroz (violin), Mauricio Freire (flute), Mauricio Loureiro (clarinet), Oilian Lana (conductor/ keyboard), and Paulo Lacerda (trombone), all UFMG faculty members.

Two reports published by the Computer Music Journal, on its 1995 summer issue, highlighted some compositions presented at the concerts (Manzolli, J. Cerana, C. 1995. "Two reports on the First Brazilian Symposium on Computer Music." Computer Music Journal. Cambridge, Massachusetts: MIT Press, 19(2):102-105.).

One of the authors, Jônatas Manzolli, from the Universidade de Campinas (UNICAMP), listed compositions that were "memorable" to him. Tape only pieces were: Barry Truax's *Sequence of Later Heaven*, *Servio Marin's Lost Villages*, Stephen Pope's *Kombination XI*, Conrado Silva's *Espaço dos Mistérios*, Flo Menezes's *La (Dé) marche sur les grains*, Fernando Lopez Lezcano's *Tree Dreams*, Francisco Kröpfl's *Mutación II*, John Chowing's *Turenas*, Jose Augusto Mannis's *Duo organum II*, Rodolfo Caesar's *Volta Redonda*, Eduardo Miranda's *Italo Calvino takes Forges Borges to a Taxi Journey in Berlin* and Aluizio Arcela's *Time Leaves*. Mixed with instrumental interaction were: Dexter Morrill's *Salzburg Variations* for trumpet and interactive system, Mauricio Loureiro's *On Behalf* for E flat clarinet and tape, and Gilberto de Carvalho's *Maelstrom* for clarinet and tape. Specially composed for the Grupo de Música Contemporânea were Sergio Freire's *Sexteto*, David Jaffe's *Impossible Animals*, and Robert Willey's *Dream Team*. "I would jump at the chance to hear any of these works again," said Manzolli.

The second author, Carlos Cerana, from LIPM, Buenos Aires, specially highlighted Dexter Morrill's *Salzburg Variations* and the collective composition by the GMC ensemble, *Os Sonhos de Little Boy*, in which "... performers played their instruments while moving about the stage in a precise interaction among instrumental gestures, displacements, and tape sounds." A similar approach was developed in Robert Willey's *Dream Team*, "... where musicians interacted with real-time computer processing of the sound." Carlos Cerana concludes his report: "Good memories of the first symposium remain with me; the high quality of the presentations, the happiness of meeting colleagues from all over the world, the gigantic work of Mauricio Loureiro (who was capable not only of coordinating an international event but also of playing the clarinet in several concerts), and the marvelous kindness of the Brazilian people. We anxiously await a similar experience at the second symposium, being scheduled for 1995. Saudades do Brasil!"

3 Computer Music as an academic research field in Brazil

Since its creation, NUCOM has always understood that there is no way to generate relevant products in computer music without the participation of highly qualified musicians and scientists. Despite having solidified itself as a subgroup of the Brazilian Computer Science Society SBC, a community consisting only of computer academics, NUCOM, at its first symposium in 1994 in Caxambu, sought to bring together researchers and artists from both music and computer science, differing from the other Special Committees of SBC for its intrinsically interdisciplinary character. The vast majority of NUCOM members were researchers and academic artists from most fields of music, such as composition, performance, musicology, and music education. However, the fact that NUCOM was institutionalized within a computer science organization seemed to have justified a tendency to define NUCOM's profile according to computer science standards.

The pertinence of an eventual institutional circumscription of the field of computer music into the music sciences became a relevant issue. Should we propose to create a new subarea of music that would stand side by side with composition, performance, musicology, and music education? Certainly not, since computer music has already been able to interweave all major fields of music. Alternatively, it should be stimulated that computer music production in Brazil would participate in the meetings of the music research community, such as the annual congresses of ANPPOM, the National Association for Research and Graduate Studies on Music. Special topic would be created to welcome works dedicated to the field. This should facilitate that computer music interdisciplinary studies continue to infiltrate into different topics of musical sciences. An important panel discussion on computer music, happened at the 1997 ANPPOM annual congress, held at the Federal University of Goiás, which gathered academic computer music researchers from music departments of universities all over the country.

Back as the general chair of the 5th edition of SBCM, held at the UFMG, in Belo Horizonte, in August 1998, I proposed a round table among NUCOM members to discuss the profile the group would or should pursue in the face of issues of academic evaluation, and the perspectives of funding for research and graduate programs in computer music. Titled, *NUCOM and Computer Music Research in Brazil: history and perspectives*, the round table was held within the official program of the V SBCM, on August 3rd, 1998. It was chaired by Bernadete Zagonel (Department of Music, UFPR). Participants were: Aluizio Arcela (Department of Computer Sciences, UnB); Rodolfo Caesar (School of Music, UFRJ); Maurício Loureiro (School of Music, UFMG); Geber Ramalho (Department of Computer Sciences, UFPE); Conrado Silva (Department of Music, UnB). Below is the report presented by the chair, Bernadete Zagonel.

3.1 Round table: NUCOM and Computer Music Research in Brazil: history and perspectives

The table began with Conrado Silva's exhibition listing the wide range of topics that have been addressed internationally in meetings occurred in this area, namely: aesthetics, artificial intelligence, equipment (audio), signal processing, composition, history of electroacoustic music, recognition audio and music, MIDI and applications, analysis, music and the brain, music education, music grammar, music languages, notation, interfaces with interpretation, psychoacoustics, scales and tuning, sound for multimedia, sound synthesis methods.

Then Mauricio Loureiro did a retrospective of the V Brazilian Symposium on Computer Music and NUCOM. He clarified the impact of the creation of the Special Commission of Computer Music within the SBC, which also included music researchers and artists from the academic community. The committee facilitated the continuation of the computer music symposia and contributed to the establishment of the area as such.

Completing these ideas, Rodolfo Caesar warned the community not to lose sight of the fact that the target element of work is music and not computing, and there is a need for more work focused on aesthetics. He brought to the forefront the problem of product evaluations by research agencies that mostly relate to the number of articles published in English on indexed journals. He then questioned an eventual academic evaluation in the area of music that might eventually exclude the production of composition or interpretation.

It was followed by Aluizio Arcela, raising the same problems about evaluation, and questioning about the types of production effectively recognized by academic institutions. Arcela sparked some controversy by asking if Computing and Music were not just serving as a garnish ("the cherry on top," or "the icing on the cake") for the SBC Congress. He noted that it would be necessary for NUCOM production to be relevant by standards of the academic computer science community.

Finally spoke Geber Ramalho, the legal representative of NUCOM at SBC at the time. He emphasized the question of the relationship between computing and music, and the interdisciplinarity that it suggests in its essence. He pointed to the seriousness and relevance of the scientific production presented at the symposia until then, highlighting the incidence of works on electroacoustic music. He identified electroacoustic music as aesthetic, while computer music would be within the technology area, covering segments, such as music education and musical analysis, by providing work tools: technology at the service of art. He raised concerns about funding, noting that all the symposia have so far been funded by the computer science area. In order to justify this, we should increase the production of computer work, a statement he has made from musicians (which were the majority), annoyance reactions, especially his argument that "all the symposia have so far been funded by the computer science area," which is far from being correct.

I hope this text may illustrate the successful trajectory traced by the computer music community emerged in this country and its contribution to the production and exposure of Brazilian art and science.

4 Acknowledgements

The organization of the first edition of the Brazilian Symposium of Computer Music, held in Caxambu, in 1994 and its fifth edition held at UFMG, in Belo Horizonte in 1998, both chaired by me, as well as the tenth edition, held at the Catholic University of Minas Gerais (PUC-MG), in Belo Horizonte, in 2005, chaired by Hugo Bastos de Paula (PUC-MG) and me, were made possible with the financial support partially provided by three research and graduate funding agencies, the National Council for Scientific and Technological Development – CNPq, the Coordination for the Improvement of Higher Education Personnel – CAPES and the Foundation for Research of the State of Minas Gerais - FAPEMIG.

The Politics of Computer Music

Fernando Iazzetta [1] *

[1] NuSom - Núcleo de Pesquisas em Sonologia – Departamento de Música da Universidade de São Paulo
Av. Prof. Lúcio Martins Rodrigues, 443 - Butantã, São Paulo - SP, 05508-020

iazzetta@usp.br

Abstract

When a set of objects, actions, and procedures begin to coalesce and gain some coherence, they become perceived as a new, cohesive field. This may be related to the emergence of a new discipline, a new craft, or a new technological configuration. As this new field shows some coherence and unity, we tend to overlook the conditions that gave rise to it. These conditions become "naturalized" as if they were inherent in that field. From this point on, we do not wonder anymore to what extent the contingencies (formal, social, economic, technological, aesthetic, religious) that gave rise to that field have been crucial to its constitution.

When it comes to computer music we are comfortably used to its applied perspective: tools, logical models, and algorithms are created to solve problems without questioning the (non-computational) origin of these problems or the directions taken by the solutions we give to them. The idea of computing as a set of abstract machines often hides the various aspects of the sonic cultures that are at play when we develop tools and models in computer music. The way we connect the development of computer tools with the contingencies and contexts in which these tools are used is what I call the *politics of computer music*. This connection is often overshadowed in the development of computer music. However, I would like to argue that this connection is behind everything we do in terms of computer music to the point that it often guides the research, development, and results within the field.

1. General Information: Full papers

First of all, I'd like to thank the organization of the SBCM for the invitation to deliver this talk. Having had the opportunity to participate in the first edition of this conference back in 1994, I am honored to be here, 25 years later, to share some ideas with this computer music community.

Before I start I want to make a warning. This text do not reflect the point of view of a computer scientist, but the one made by an artist and teacher who has developed a close relationship with computer music technology. Taken from this perspective, this text could be fairly titled: How I Understand Computer Music and What I Would Like It to Do For Me.

* Supported by CNPq, proc. # 309143/2018-7.

I want to begin talk by addressing the two pictures below. They both refer to the use of technology in works that are part of our artistic production at NuSom, the Research Center on Sonology at the University of São Paulo.

Figure 1: Photo from *¿Música?* 2 (2009)

Figure 2: Photo from *¿Musica?* 14 (2019)

Both pictures are taken from performances produced by NuSom in a series os events called ¿Musica?. The series was created as a way to provoke the discussion about academic practices at the Department of Music at the University of São Paulo, which, like most music departments in the country, focuses on concert music from the European tradition. The series started in 2006 and currently is in its 16th edition.

What connects these two images is not the use of digital technologies in a music performance, nor the fact that they show two video projectors. What interests me about them is how they can reveal the context in which they were used.

The first image is somehow explicit: a video projector tied with strings is not just an unusual technical solution. It indicates the coexistence between hi and low-tech, between sophistication and the *gambiarra*.

The second refers to a concert we gave last April at the Planetarium of São Paulo and may require some explanation. The choice of the Planetarium to perform that concert was not only aesthetic, but also political. This is a remarkable facility, opened in 1957, which is part of São Paulo's cultural life. But in recent years it only works twice a week because it relays on a rather reduced staff and on a very small budget. In this constraint scenario the Planetarium cannot maintain a more intense regular schedule. Therefore, choosing this site to present the performance had more to do with giving visibility to it than to the fact that it offers and amazing and sophisticated setup of video projectors.

I began by bringing these two examples because they refer to concrete technological problems, even though these problems do not concern how these machines work nor the techniques that are involved in their operation.

This kind of circumstance constantly induces and regulates the actions and decisions we make when we create and use a technology, but we rarely think of these issues as technological problems in themselves. However, I can only understand technology as a device that acts on something and has an agency. We do things with them. But to a large extent they also make us to be the way we are and to act the way we do.

2. Computer Music

There is nothing new about this approach and it may be slightly related to what has been called material culture studies: the study of usage, consumption, creation, and trade of objects as well as the behaviors, norms, and rituals that the objects create or take part in.

When a set of objects, actions, and procedures begin to coalesce and gain some coherence, they become perceived as a new, cohesive field. This may be related to the emergence of a new discipline, a new craft, or a new technological configuration. As this new field shows some coherence and unity, we tend to overlook the conditions that gave rise to it. These conditions become "naturalized" as if they were inherent in that field. From this point on, we do not wonder anymore to what extent the contingencies (formal, social, economic, technological, aesthetic, religious) that gave rise to that field have been crucial to its constitution.

Of course, as the field develops in time we have the impression that its concerns, methods and tools change accordingly. It seems that there is a direct relation between the emergence of new problems and the development of new solutions (techniques, technologies, methods).

When it comes to computer music we are comfortably used to its applied perspective: tools, logical models, and algorithms are created to solve problems without questioning the (non-computational) origin of these problems or the directions we take when finding solutions to them. The idea of computing as a set of abstract machines often hides the various aspects of the sonic cultures that are at play when we develop tools and models in computer music.

The way we connect the development of computer tools with the contingencies and contexts in which these tools are used is what I call the politics of computer music. This connection is often overshadowed in the development of computer music. However, I would like to argue that this connection is behind everything we do in terms of computer music to the point that it often guides the research, development, and results within the field.

When we disregard this connection (this politics) we pretend that our choices are neutral in terms of ethics and, particularly in the case of computer music, in terms of aesthetics as well.

In this text, I would like to consider the politics of computer music as a way to critically explore the field. After more than 2 decades working in this area which is loosely called "music and technology" – and technology here is often taken as a synonym with computing – it seems to me that there is no point in thinking computer music without thinking its politics. This is obviously a personal position, which I would like to share with the reader, without it implying any disregard for other forms of action in the field.

I regard technology not only as a tool we use to accomplish something. A technology is a way of thinking. It contains in its modes of operation a frame of what will be done and how it will be done. Roland Barthes [1], referring to Russian linguist Roman Jákobson, has already said that language – one of our most fundamental technologies – is defined much more by what it forces us to say than by what it allows us to say. Instruments, software and machines are not passive means. As Thomas Hankins and Robert Silverman point in their book about instruments and the imagination, "Instruments have a live of their own. They do not merely follow theory; often they determine theory, because instruments determine what is possible, and what is possible determines to a large extent what can be taught". [2]

When I choose to use or produce a tool I am choosing a way of thinking (aesthetics, poetics) and a way of acting (ethics, politics). Therefore tools simultaneously occupy and configure a place and a time. They set up a context, but the context also determines what technology is, the agency it produces.

History is full of examples of how certain technologies acquire new perspectives when changing the context in which they are used. Thomas Edson could not foresee the use of his phonograph as a music tool when he wrote the device's patent 1878. But, by the end of the nineteenth century the phonograph was already thought of as a musical instrument.

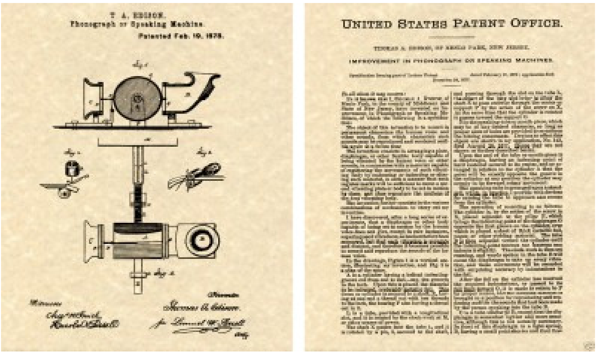


Figure 3: Thomas Edison's 1877 Phonograph Patent

Another prominent inventor, the Brazilian Santos Dumont, got depressed when who saw his invention, the airplane, been used to kill people in the First World War instead of helping to connect them.

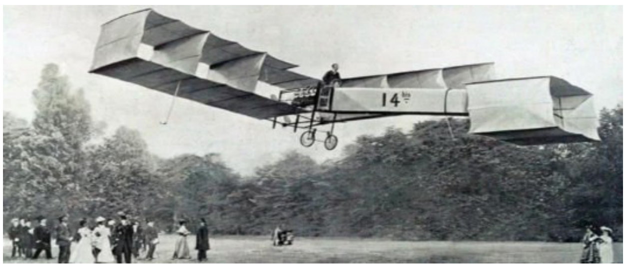


Figure 4: Santos Dumont's famous airplane 14 Bis

The same could be said about the profound discomfort of Christopher Cerf, the musician who composed about 200 songs for the educational TV show Sesame Street when he discovered that his children's songs were used as one of the most terrible forms of torture in Guantanamo Prison.

Figure 5: Sesame Street Show

Figure 6: Guantanamo Prison

3. The SBCM

So lets try to bring the discussion to the local context of Brazilian computer music and take a look at some aspects that influenced the field in Brazil through the critical lenses of its politics.

In 1994 a friend of mine, professor Mauricio Loureiro, suggested me to submit a work to a computer music conference he was organizing. It was the first edition of this conference, the Brazilian Symposium on Computer Music. Looking back at it I now realize how remarkable this meeting was. It gathered some well-known names of the field such as Francisco Kropfl, David Jaffe, Xavier Serra and Stephen Travis Pope in a very intense meeting. Although there were many people involved in this endeavor, I must credit the success of the event to its Chair, Mauricio Loureiro.

I understand that this first SBCM represented the beginning of a systematic work on computer music in Brasil. As one might expect, this early phase was based on models inspired by other major research centers in the area such as IRCAM, CRRMA, or the Institute of Sonology (The Hague).

In fact, to some extent SBCM mirrored itself to the structure of similar conferences, especially the ICMC. This initial effort was undoubtedly one of the factors that contributed to the growth of computer music in Brazil and enabled the exchange with researchers and groups from other countries, specially from Latin America and the United States. Indeed, it would be difficult to think of a field of computer music today that would remain outside of a global perspective that involves not only sharing common problems and technologies, but also moving through a network of conferences, publications, exchanges between research groups and eventually sharing funding.

On the other hand, I understand that this adherence to these existing models should be counterbalanced by an investigation of the particularities of the Brazilian socio-cultural context.

4. The Musarts

I just want to give an example of how decisive the political and social environment can be when we decide to develop something around technology. In 2003, the scientific director of the São Paulo Research Foundation (Fapesp) had the opportunity to visit Ircam during a trip to France. He was impressed by the connection between art, science, and technology promoted at the center and upon returning to Brazil, he proposed to a group of scholars from São Paulo the creation of a research center inspired by the French institute. Despite the significant differences between the cultural and economic realities of Brazil and France, the technocentric and scientificist model taken as representative of Ircam's mode of action seemed extremely appealing to Fapesp.

As one can imagine, it is not every day that an agency like Fapesp decides to allocate to a field such as computer music resources similar to those it allocates to the study of cancer treatment or to support the construction of communication satellites. This Brazilian Virtual Institute of Music and Technology was named MusArtS - Musica Articulata Scientia - but for various reasons, including lack of consensus among the researchers involved in the project, it was never implemented. But I suspect that Musarts did not succeed just because we didn't take into account the remarkable differences between French and Brazilian environments.

4. Computer Politics

Besides social cultural and economic differences, the way computers were introduced in Brazil occurred in a very particular way (this topic has been developed in [3]. From the 1990s, two events are significant to an understanding of the expansion of the area of music and technology in the country.1 The first concerns a series of restrictions on imports of various items that were not manufactured in the country. These restrictions were supported by market reserve policies that were intended to protect the local industry from competition from developed countries. These restrictions had been applied since *Estado Novo* in the 1930s, but were intensified during the military dictatorship in the 1960s and 1970s. In spite of the rich Brazilian musical production, especially in the field of popular music, access to imported audio equipment and instruments (acoustic or electronic) was very restricted, which generated an informal (and illegal) market of considerable dimensions to feed the demand from this local music industry.

The *Política Nacional de Informática* (PNI [National Informatics Policy]) was established in 1984, also with the idea of stimulating the development of the computer industry in Brazil by forcing a market reserve for national capital companies. In general, its effects were not as expected: the sector developed poorly and consumers were forced to pay exorbitant prices for obsolete products compared to the technologies available in other countries. In practice, Law No. 7,234/84, which reserved for domestic capital manufacturers the right to produce and sell computer goods, did not last long. Unable to meet the demand for computers and programs, the market reserve was revised in 1991 and abolished in 1992. But, in its eight years it paved the way for the creation of countless Brazilian companies, leaving behind a trail of controversy.

From the 1990s, access to digital technology was gradually established, and that had a direct impact on the emergence of new studios focused on computer music production. This process coincided with the increasing digitalization of musical and sound production, making audio technologies significantly cheaper. In the following decade, significant investments were made to consolidate digital inclusion policies. Pontos de Cultura (PdC [Points of Culture]) was an initiative of the Ministry of Culture in the government of Luiz Inácio Lula da Silva that reflected a series of changes in the prospects of cultural production fostering. The PdC aimed to provide direct funding to community groups and non-governmental entities to develop activities in local communities. There was not one single model, in order to avoid the homogenization of actions that extended to quite different sectors and contexts. Based on the ideas of autonomy and empowerment, the program exposed its clearly Marxist foundations to promote an increase in the cultural production of a sector of the population that was historically disenfranchised from institutional support mechanisms. From indigenous communities in remote areas to organized groups in the large urban centers of the country, a network was established whose main characteristic was, perhaps, diversity. It is important to note that part of the funds allocated to each PdC project was destined for purchasing a multimedia station (a mini audio-recording studio, a small video-editing station, and computers using free software). Multiplied by the thousands, the PdC projects provided access for an invaluable number of young people from different social strata to audiovisual production equipment, thus increasing production, especially in the less-favored sectors of Brazilian society. With that, a whole generation of artists formed outside the commercial circuits and distant from the support of the academy, to become a protagonist in the arts field. The impact of this cultural effervescence reverberates in the current period with the formation of a rich experimental scene that attracted little attention from academic research in computer music. Again, more global concerns seemed to guide our attention.

5. The NuSom

I would like take this opportunity to briefly present the work being done by NuSom in recent years. The Center houses several groups that operate with some autonomy in different fields. Some are more linked to artistic practice, others to theoretical reflection and still others to technological development.

For the last 15 years we have produced a considerable amount of artistic and academic work in which the use of computational technologies is quite significant. A central question for the group's work is how to make artistic production, academic research and technological development compatible. Several projects were developed in this sense, in collective and individual works.

Particularly in the *¿Música?* Series (see [4] and [5]) there is a strong concern about the employment and the development of technologies when they are taken into the artistic process which often implies a questioning regarding a positivist and scientific perspective of the use of these technologies. Similarly, the exploration of unusual spaces for artistic practice (*¿Música?* 6), or the incorporation of scenic, visual and textual elements (*¿Música?* 1 and 2), are always accompanied by critical discussions about these modes of production. For this reason, several works are built around themes that are discussed and studied in advance by the group. These themes may vary from abstract concepts - such as political militancy (*¿Música?* 12 and 13); the use of silence as musical material (*¿Música?* 11); mediation between the visual and the sound domain (*¿Música?* 3); or the exploration of improvisation techniques (*¿Música?* 8). There is constant care with the registration of these works, which will be discussed after they are submitted. These discussions, in turn, will feed the group's upcoming productions. That is, we seek to build a circular process in which research feeds artistic production and artistic production is taken as material for academic research itself.

This feedback between research, creation and reflection implies a process of constant evaluation of the role of technologies within the group's production. Computational tools are not taken as neutral elements, but as critical elements of the creative processes. In fact, the realization of artistic works is not taken as a neutral action that is referred only to the art world (art for the art sake), but as an aesthetic action that implies a political expression. Thus, the use of technologies, as a significant part of the group's poetics, is also subjected to a reflection on its ethical and aesthetic consequences. I take Nuson's artistic production not because it should be taken as an exemplary model of how one should deal with the political aspect of technology. On the contrary, my intention is to point out some of the tensions and contradictions that are at stake when we use music technologies without taking into account the contexts, the stories, and the contingencies in which they were created. The risk is that they rule the creative process and hide our own stories and our own aesthetic thinking.

References

[1] Barthes, Roland (s/d). *Aula*. São Paulo: Cultrix [originaly published in French in 1978].

[2] Hankins, Thomas & Robert Silverman (1995). *Instruments and the Imagination*. New Jersey: Princeton University Press.

[3] Iazzetta, Fernando (2019). “Sounds from Elsewhere: Episodes for a History of Brazilian Sound Art.” In *Making it Heard: a History of Brazilian Sound Art* (R. Chaves & F. Iazzetta, Eds.). New York: Bloomsburry, 12-42.

[4] Iazzetta, Fernando & Lílian Campesato (2019). “La Serie ¿Música? Arte sonoro desde contextos locales”. *I Simposio Internacional de Arte Sonoro: Mundos Sonoros: cruces, circulaciones, experiencias*. Buenos Aires: Untref, 143-152.

[5] Kisil, Vitor (2014). *¿Música?*: processos e práticas de criação e performance em um ambiente de pesquisas em sonologia, PhD Thesis, Departamento de Música, Universidad de São Paulo.

Authors

A
Adilson Siqueira, 231
Alan Taitz, 197
Alexandre Porres, 121
Aluizio Oliveira, 165, 233
Ana Julia Pereira Caetano, 58
Ana Miccolis, 12
André Dalmora, 17
Andre Du Bois, 106
André Gomes, 114, 157, 227
Avner Paulo, 28, 157, 227, 231

B
Bruna Guimarães Lima e Silva, 231
Bruno Gadelha, 42
Bruno Masiero, 155
Bruno Mesz, 197

C
Carlos Almada, 12
Carlos Eduardo Oliveira de Souza, 231
Carlos Soares Araujo, 141
Carlos Thomaz, 194
Christhian Fonseca, 125
Claudia Usai, 12
Claudio Gomes, 210
Clayton Mamedes, 92
Cristiano Figueiró, 35

D
Daniel Coura, 151
Danilo Rossetti, 4
Davi Mota, 165
Diego Shalom, 197

E
Eduardo Cabral, 12
Eduardo Magalhães, 204
Edward David Moreno, 50
Erivelton Geraldo Nepomuceno, 204
Esteban Viveros Astorga, 35
Estela Ribeiro, 194

F
Fabiano Costa, 151
Fabio Carvalho, 225
Felippe Barros, 22
Fernando Iazzetta, 35, 248
Filipe Calegario, 76, 173, 217
Flávio Schiavoni, 28, 63, 69, 99, 114, 157, 201, 225, 227, 231
Frederico Resende, 114, 157, 227

G
Gabriel e Silva, 92
Gabriel Lopes Rocha, 69, 157, 227
Geber Ramalho, 76, 173, 217
Giordano Cabral, 76, 173, 217

H
Helder Rocha, 151

I
Igino Silva Junior, 28, 63, 157, 227
Igor Andrade, 157, 227
Igor Chagas, 12

J
João Monnazzi, 229
João Penchel, 12
João Teixeira, 28, 69, 157, 201, 225, 227
João Tragtenberg, 76, 217
Jonatas Manzolli, 4
José Eduardo Ayres, 223, 239
José Ezequiel Sánchez, 223
Josue da Silva, 210
Julian J. Arango, 35
Juliano Foleiss, 131

L
Leandro Costalonga, 84, 151
Leonardo Brum, 50
Leonardo Ferreira, 194
Luan Gonçalves, 99, 114
Lucas Marques, 136
Luiz Velho, 239
Luzilei Aliel, 221

M
Marcelo Johann, 181
Marcelo Pimenta, 84, 181
Marcelo Queiroz, 212
Marco Cristo, 141
Marco Leal, 210
Marcos Trevisan, 197
Marcus Vinícius Neves, 151
Maurício Loureiro, 22, 165, 199, 243
Mauricio Perez, 207
Mauro Amazonas, 42
Mauro César Fachina Canito, 28, 157
Max Kuhn, 12
Micael Antunes da Silva, 4

O
Octávio Deluchi, 201

P
Paulo Vitor Itaboraí de Barros, 189
Pedro Arthur, 223, 239

R
Rafael Fajiolli, 221
Rafael Giusti, 141
Regis Rossi Alves Faria, 207, 229
Ricardo Thomasi, 221
Rodolfo Coelho de Souza, 207
Rodrigo Ribeiro, 106
Rodrigo Schramm, 181
Rogerio Constante, 201
Rômulo Augusto Vieira Costa, 28, 157
Ronaldo Prati, 215
Rosiane de Freitas, 42

S
Shayenne Moura, 212

T
Tairone Magalhaes, 22, 165
Thais Castro, 42
Thiago Campolina, 165
Thiago Nascimento, 210
Thiago Santos, 204
Tiago Tavares, 17, 58, 125, 131, 155, 237

V
Vinicius Braga, 12
Vitor Rolla, 223, 239
Vítor Shinohara, 131

W
Willy Cornelissen, 199

Y
Yubiry González, 215

Institutions

C
Caldas University - Colombia, 35

F
Federal Technological University of Paraná - Brazil, 131
Federal University of ABC - Brazil, 215
Federal University of Alfenas - Brazil, 136
Federal University of Amapá - Brazil, 210
Federal University of Amazonas - Brazil, 42, 141
Federal University of Espirito Santo - Brazil, 84, 151
Federal University of Espírito Santo - Brazil, 151
Federal University of Minas Gerais - Brazil, 22, 165, 199, 233, 243
Federal University of Ouro Preto - Brazil, 106
Federal University of Paraná - Brazil, 92
Federal University of Pelotas - Brazil, 106, 201
Federal University of Pernambuco - Brazil, 76, 173, 217
Federal University of Rio de Janeiro - Brazil, 12
Federal University of Rio Grande do Sul - Brazil, 84, 181
Federal University of São João del-Rei - Brazil, 28, 63, 69, 99, 114, 157, 201, 204, 225, 227, 231
Federal University of São Paulo - Brazil, 35, 221
Federal University of Sergipe - Brazil, 50

N
National Institute of Pure and Applied Mathematics - Brazil, 223, 239
National University of Tres de Febrero - Argentina, 197

S
SENAI Innovation Institute for ICT - Brazil, 76, 173, 217
State University of Campinas - Brazil, 58, 125, 131, 155, 237

U
Universidade Federal de São João del-Rei - Brazil, 201
University Center of FEI - Brazil, 194
University of Buenos Aires - Argentina, 197
University of Campinas - Brazil, 4, 17, 131
University of São Paulo - Brazil, 121, 189, 207, 212, 229, 248

Keywords

A
Acoustics, Diffusion and Sonorization, 225
Artificial Intelligence, A-Life and Evolutionary Music Systems, 92, 199, 210

B
Brain-Computer Interfaces and Physiological Signals, 194

C
Computational Musicology, 17, 151, 197, 201, 227
Computer Music and Creative process, 28, 35, 42, 92, 151, 155, 157, 173, 201, 217, 221, 223, 225, 231, 233, 239

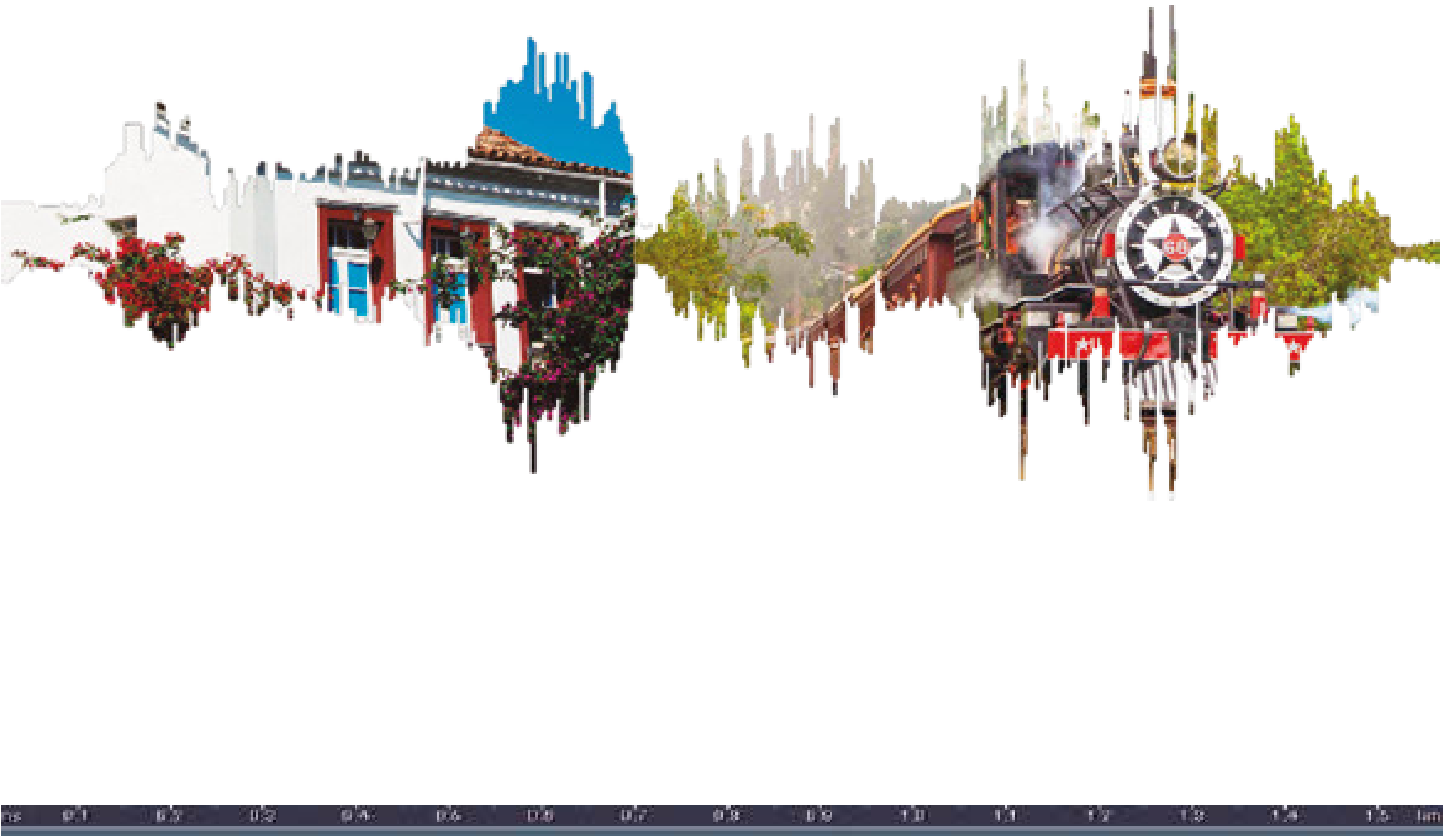
D
Digital Music Systems and Services, 69, 157
Digital Sound Processing, 99, 151, 181, 189, 204, 207, 212, 215

I
Internet and Web Applications, 28

M
Movement and Gesture, 58, 69, 165, 227, 229
Multimedia Systems, 35, 173, 217, 221
Music Analysis and Synthesis, 4, 12, 50, 92, 136, 141, 199, 212, 215, 233, 237
Music Expressiveness, 22, 42, 106, 165, 223
Music Formats, Data Structures and Representation, 35, 106
Music Information Retrieval, 4, 17, 22, 125, 131, 136, 141, 155, 181, 189, 194, 199, 212, 237
Music Notation, Printing, and Optical Recognition, 201
Music Perception, Psychoacoustics and Cognition, 4, 84, 121, 194, 197
Music, Emotion and Communication, 63, 210, 231
Music, Society and Technology, 28, 63, 76, 84, 99, 173, 225, 227, 231

R
Real-time Interactive Systems, 42, 50, 58, 76, 121, 210, 217, 221, 223, 229

S
Sensors and Multimodal Signal Processing, 58, 76, 189, 229
Software Systems and Languages for Sound and Music, 22, 50, 99, 106, 114, 157, 165, 239
Sound Analysis and Synthesis, 121, 215, 233



Acknowledgments:

UFSJ

PPGCC
Programa de Pós-Graduação em Ciência da Computação

Prope

PIPAUS

PROEX UFSJ
Pró-Reitoria de Extensão e Assuntos Comunitários

CAPES

SBC
Sociedade Brasileira de Computação

CNPq